

*Trabajo Fin de Grado*

“CONTROL SYSTEM FOR AUTOLANDING IN NON-  
FULLY HORIZONTAL SURFACES”

---

Alejandro Montoya Sánchez de la Morena

Tutor

Abdulla Hussein Abdulrahman Al Kaff

Leganés, 8 /10/ 2018



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento – No Comercial – Sin Obra Derivada**



## **AGRADECIMIENTOS**

Me gustaría dedicar este proyecto, en primer lugar, a mi familia, especialmente a mis padres, mi hermana, mi tío Juan, Mari Carmen y Lourdes por su apoyo y su cariño incondicional durante toda mi etapa académica, por haberme animado y aguantado en los peores momentos, por haberme brindado las oportunidades que me han brindado y por haberme formado como persona, gracias a ellos he llegado hasta aquí.

También me gustaría dedicarles este trabajo a mis amigos por hacerme más llevaderos estos meses y por hacer cada año mejor que el anterior.

Por supuesto, me gustaría agradecerle a mi tutor la oportunidad de realizar este proyecto, al personal del laboratorio de sistemas inteligentes de la Carlos III y, en especial, a Ricardo por su infinita paciencia. Por último, un agradecimiento a Emanuel Koslosky que me resolvió muchísimas dudas sobre su programa y el funcionamiento de la lógica difusa.

Gracias





## RESUMEN

El objetivo principal de este proyecto es la creación de un controlador basado en lógica difusa para el aterrizaje autónomo en superficies que puedan presentar cierta inclinación. Además, se compara la respuesta frente a distintas combinaciones de funciones de membresía y se evalúan los resultados.

En este estudio se han evaluado tres posibles controladores: PID, LQR y Fuzzy Logic; resultando seleccionado este último por su versatilidad y sencillez. Adicionalmente, se han realizado una serie de pruebas con distintos escenarios y de los resultados obtenidos en las simulaciones se deduce que en ausencia de inclinación o cuando solo existe rotación respecto a un eje, el controlador responde de manera satisfactoria. Sin embargo, en el último escenario, en el cual se presenta rotación respecto a dos ejes, presenta algunos fallos por lo que en futuras investigaciones sería necesario la implementación de un sistema de optimización para subsanarlos.

**Palabras clave:** Dron; Control Difuso; Aterrizaje; Autónomo; No Lineal.

## ABSTRACT

The main objective of this project is the creation of a controller based on fuzzy logic for autonomous landing on surfaces that may have a certain inclination. In addition, the response is compared to different combinations of membership functions and the results are evaluated.

In this study, three possible controllers have been evaluated: PID, LQR and Fuzzy Logic; the latter being selected for its versatility and simplicity. Additionally, a series of tests have been carried out with different scenarios and from the results obtained in the simulations it can be deduced that in the absence of inclination or when there is only rotation with respect to an axis, the controller responds satisfactorily. However, in the last scenario, in which rotation is presented with respect to two axes, it presents some failures, so in future investigations it would be necessary to implement an optimization system to correct them.

**Keywords:** UAVs; Fuzzy Control; Landing; Autonomous; Non-linear.

# CONTENIDO

<b>CAPÍTULO 1. INTRODUCCIÓN.....</b>	<b>1</b>
1.1 Motivación.....	4
1.2 Problema.....	4
1.3 Solución que se propone, Fuzzy Logic Controller (FLC) o controlador de lógica difusa:.....	4
<b>CAPÍTULO 2. ESTADO DEL ARTE .....</b>	<b>6</b>
2.1 Controlador clásico PID .....	6
2.2 Controlador LQR.....	9
2.3 Controlador basado en lógica difusa .....	11
<b>CAPÍTULO 3. ALGORITMO PROPUESTO .....</b>	<b>17</b>
3.1 Ecuaciones.....	17
3.2 Modo de trabajo para diseñar un controlador .....	20
3.3 Algoritmo del controlador .....	22
3.4 Librerías lógica difusa .....	24
<b>CAPÍTULO 4. RESULTADOS EXPERIMENTALES .....</b>	<b>25</b>
4.1 Plataformas.....	25
4.1.1 ROS.....	25
4.1.2 Ar_track_alvar.....	26
4.1.3 MAVROS.....	27
4.1.4 MAVLink.....	28
4.1.5 PIXHAWK PX4.....	29
4.2 Simulaciones.....	33
4.2.1 Comparación entre funciones de membresía trapezoidal y triangular para los ejes X y Z: .....	34
4.2.2 Comparación entre funciones de membresía trapezoidal y triangular para el eje Y: .....	41
4.2.3 Comparación entre funciones de membresía trapezoidal y triangular para el ángulo $\alpha$ :.....	47
4.2.4 Comparación entre funciones de membresía trapezoidal y triangular para el ángulo $\beta$ :.....	56
4.2.5 Análisis de la respuesta de los parámetros X, Y y Z sin variación angular: .....	65
4.2.6 Análisis de la respuesta de los parámetros X, Y y Z en presencia de variación angular $\alpha$ : .....	69
4.2.7 Análisis de la respuesta de los parámetros X, Y y Z en presencia de variación angular $\beta$ : .....	73
4.2.8 Análisis de la respuesta de los parámetros X, Y y Z en presencia de variación angular $\alpha$ y $\beta$ : .....	77
<b>CAPÍTULO 5. CONCLUSIONES Y TRABAJOS FUTUROS .....</b>	<b>81</b>
5.1 Conclusiones: .....	81
5.2 Trabajos futuros: .....	82
<b>BIBLIOGRAFÍA .....</b>	<b>84</b>



## Índice de figuras

Fig. 1.1 Ejemplo dron ala fija [2] .....	1
Fig. 1.2 Ejemplo dron de alas de aleteo con forma de murciélago [3] .....	2
Fig. 1.3 Ejemplo dron dirigible [4].....	2
Fig. 1.4 Ejemplo dron hexacopter [7].....	3
Fig. 1.5 Posibles combinaciones de las hélices [8].....	3
Fig. 2.1 Gráfico comparativo entre conjunto difuso y conjunto lógico clásico [12] .....	12
Fig. 2.2 Comparativa entre los distintos métodos del proceso de 'Defuzzification' [15].....	16
Fig. 2.3 Esquema genérico de un sistema basado en lógica difusa [16].....	16
Fig. 3.1 Plano ZY .....	18
Fig. 3.2 Plano ZX .....	19
Fig. 3.3 Workflow para diseñar un sistema de lógica difusa [9].....	21
Fig. 3.4 Representación del sistema de control de lógica difusa [9]. .....	22
Fig. 3.5 Workflow del controlador de lógica difusa incluyendo topics [6]. .....	23
Fig. 3.6 Diagrama UML de las clases de las librerías de lógica difusa [14]. .....	24
Fig. 4.1 Esquema del flujo de información de ROS [17]. .....	25
Fig. 4.2 'April Tag' combinado [9].....	26
Fig. 4.3 Esquema de MAVROS [19] .....	27
Fig. 4.4 Estructura de un mensaje típico que sigue el protocolo MAVLink [21]. .....	28
Fig. 4.5 Placa Pixhawk conectada a los periféricos [27] .....	31
Fig. 4.6 Orientación Yaw en función de la orientación de la placa [29].....	32
Fig. 4.7 Orientación de los ejes de coordenadas cuando Yaw es 0° [24].....	32
Fig. 4.8 Funciones de membresía para Z en disposición trapezoidal .....	34
Fig. 4.9 Funciones de membresía para Z en disposición triangular .....	35
Fig. 4.10 Funciones de membresía para VelZ en disposición triangular .....	35
Fig. 4.11 Funciones de membresía para VelZ en disposición trapezoidal .....	36
Fig. 4.12 Gráfico comparativo para Z y VelZ con combinación trapecio-triángulo.....	37
Fig.4.13 Gráfico comparativo para Z y VelZ con combinación triángulo-triángulo .....	38
Fig.4.14 Gráfico comparativo para Z y VelZ con combinación trapecio-trapecio .....	39
Fig.4.15 Gráfico comparativo para Z y VelZ para combinación triángulo-trapecio.....	40
Fig. 4.16 Funciones de membresía para Y con disposición trapezoidal .....	41
Fig. 4.17 Funciones de membresía para Y con disposición triangular .....	41
Fig. 4.18 Funciones de membresía para VelY con disposición triangular .....	42
Fig. 4.19 Funciones de membresía para VelY con disposición trapezoidal.....	42
Fig. 4.20 Gráfico comparativo para Y y VelY con combinación trapecio-trapecio.....	44
Fig. 4.21 Gráfico comparativo para Y y VelY con combinación trapecio-triángulo .....	45
Fig. 4.22 Gráfico comparativo para Y y VelY con combinación triángulo-triángulo.....	46
Fig. 4.23 Gráfico comparativo para Y y VelY con combinación triángulo-trapecio .....	47
Fig. 4.24 Funciones de membresía para alfa con disposición trapezoidal.....	48
Fig. 4.25 Funciones de membresía para alfa con disposición triangular.....	48
Fig. 4.26 Funciones de membresía para Wx con disposición trapezoidal .....	49
Fig. 4.27 Funciones de membresía para Wx con disposición triangular .....	49
Fig. 4.28 Gráfico de la evolución de alfa con combinación trapecio-trapecio .....	51
Fig. 4.29 Gráfico de la evolución de Wx con combinación trapecio-trapecio .....	51
Fig. 4.30 Gráfico de la evolución de alfa con combinación trapecio-triángulo.....	52
Fig. 4.31 Gráfico de la evolución de Wx con combinación trapecio-triángulo .....	53
Fig. 4.32 Gráfico de la evolución de alfa con combinación triángulo-triángulo .....	54
Fig. 4.33 Gráfico de la evolución de Wx con combinación triángulo-triángulo .....	54
Fig. 4.34 Gráfico de la evolución de alfa con combinación triángulo-trapecio.....	55
Fig. 4.35 Gráfico de la evolución de alfa con combinación triángulo-trapecio.....	56
Fig. 4.36 Funciones de membresía para beta con disposición trapezoidal.....	57
Fig. 4.37 Funciones de membresía para beta con disposición triangular .....	57
Fig. 4.38 Funciones de membresía para Wy con disposición trapezoidal .....	58

Fig. 4.39 Funciones de membresía para beta con disposición trapezoidal .....	58
Fig. 4.40 Gráfico de la evolución de beta con combinación trapecio-trapecio .....	60
Fig. 4.41 Gráfico de la evolución de Wy con combinación trapecio-trapecio.....	60
Fig. 4.42 Gráfico de la evolución de beta con combinación trapecio-triángulo.....	61
Fig. 4.43 Gráfico de la evolución de Wy con combinación trapecio-triángulo .....	62
Fig. 4.44 Gráfico de la evolución de beta con combinación triángulo-triángulo .....	63
Fig. 4.45 Gráfico de la evolución de Wy con combinación triángulo-triángulo.....	63
Fig. 4.46 Gráfico de la evolución de beta con combinación triángulo-trapecio.....	64
Fig. 4.47 Gráfico de la evolución de Wy con combinación triángulo-trapecio .....	65
Fig. 4.48 Gráfico comparativo del desplazamiento en ausencia de inclinación .....	66
Fig. 4.49 Gráfico comparativo de la velocidad en ausencia de inclinación .....	67
Fig. 4.50 Gráfico comparativo X y VelX en ausencia de inclinación.....	67
Fig. 4.51 Gráfico comparativo Y y VelY en ausencia de inclinación .....	68
Fig. 4.52 Gráfico comparativo de Z y VelZ en ausencia de inclinación.....	68
Fig. 4.53 Nuevas funciones de membresía para Z con región de rectificación .....	69
Fig. 4.54 Nuevas funciones de membresía para VelZ con región de rectificación .....	70
Fig. 4.55 Gráfico comparativo del desplazamiento en presencia de ángulo alfa.....	71
Fig. 4.56 Gráfico comparativo para las velocidades en presencia de ángulo alfa .....	71
Fig. 4.57 Gráfico comparativo de X y VelX en presencia de alfa .....	72
Fig. 4.58 Gráfico comparativo de Y y VelY en presencia de alfa.....	72
Fig. 4.59 Gráfico comparativo entre Z y VelZ en presencia de alfa .....	73
Fig. 4.60 Gráfico comparativo del desplazamiento en presencia de beta.....	74
Fig. 4.61 Gráfico comparativo de las velocidades en presencia de beta .....	75
Fig. 4.62 Gráfico comparativo de X y VelX en presencia de beta.....	75
Fig. 4.63 Gráfico comparativo de Y y VelY en presencia de beta .....	76
Fig. 4.64 Gráfico comparativo de Z y VelZ en presencia de beta.....	76
Fig. 4.65 Gráfico comparativo del desplazamiento en presencia de alfa y beta.....	78
Fig. 4.66 Gráfico comparativo de las velocidades en presencia de alfa y beta .....	78
Fig. 4.67 Gráfico comparativo de X y VelX en presencia de alfa y beta.....	79
Fig. 4.68 Gráfico comparativo de Y y VelY en presencia de alfa y beta .....	79
Fig. 4.69 Gráfico comparativo de Z y VelZ en presencia de alfa y beta.....	80

## Índice de tablas

Tabla 1 Reglas difusas para Z.....	36
Tabla 2 Datos Z y VelZ combinación trapecio-triángulo.....	37
Tabla 3 Datos Z y VelZ combinación triángulo-triángulo.....	38
Tabla 4 Datos Z y VelZ combinación trapecio-trapecio.....	39
Tabla 5 Datos Z y VelZ combinación triángulo-trapecio.....	40
Tabla 6 Reglas difusas para Y.....	43
Tabla 7 Datos Y y VelY combinación trapecio-trapecio.....	43
Tabla 8 Datos Y y VelY combinación trapecio-triángulo.....	44
Tabla 9 Datos Y y VelY combinación triángulo-triángulo.....	45
Tabla 10 Datos Y y VelY combinación triángulo-trapecio.....	46
Tabla 11 Reglas difusas alfa.....	49
Tabla 12 Datos alfa y Wx combinación trapecio-trapecio.....	50
Tabla 13 Datos alfa y Wx combinación trapecio-triángulo.....	52
Tabla 14 Datos alfa y Wx combinación triángulo-triángulo.....	53
Tabla 15 Datos alfa y Wx combinación triángulo-trapecio.....	55
Tabla 16 Reglas difusas beta.....	58
Tabla 17 Datos beta y Wy combinación trapecio-trapecio.....	59
Tabla 18 Datos beta y Wy combinación trapecio-triángulo.....	61
Tabla 19 Datos beta y Wy combinación triángulo-triángulo.....	62
Tabla 20 Datos beta y Wy combinación triángulo-trapecio.....	64
Tabla 21 Datos X, Y y Z en ausencia de inclinación.....	66
Tabla 22 Datos X, Y, Z y alfa en ausencia de beta.....	70
Tabla 23 Datos X, Y, Z, beta en ausencia de alfa.....	74
Tabla 24 Datos X, Y, Z, alfa y beta.....	77

## CAPÍTULO 1. INTRODUCCIÓN

En la actualidad, el mundo avanza hacia la automatización y el área de los vehículos no es una excepción. Es muy posible que llegue el día en que conducir se haga extraño, pero de momento llevan la delantera los vehículos aéreos no tripulados o UAV en inglés (*Unmanned Aerial Vehicle*) ello puede ser debido a que resultan más fáciles de manipular, conllevan menos riesgo y puedan abarcar un amplio espectro de tareas que antes resultaban más tediosas.

Su potencial es enorme, hoy en día no es difícil encontrar drones tomando fotos, grabando todo tipo de eventos o decorándolos como en la Superbowl 2017; además, también se han implementado en estudios topológicos, ayudando a las administraciones públicas a realizar controles más exhaustivos sobre las propiedades rústicas y urbanas. Adicionalmente, se están utilizando en tareas de rescate, en el control de incendios forestales, en vigilancia fronteriza, el envío de paquetes, la manipulación de objetos peligrosos, el control de rebaños, el estudio de comportamientos biológicos de los animales y como no, divirtiéndose a la gente con un fin puramente lúdico [1].

Hay muchos tipos de vehículos no tripulados y se pueden clasificar de muchas formas, la más común es en función de su modo de vuelo, así pues, podemos encontrar las siguientes categorías:

- **Fixed-wing o ala fija:** Este tipo de drones son los que más se parecen a los aviones convencionales. Son capaces de alcanzar altas velocidades y necesitan una pista para aterrizar y despegar



Fig. 1.1 Ejemplo dron ala fija [2]

- **Flapping-wing o alas de aleteo:** estos drones suelen ser pequeños y de alas flexibles que imitan el aleteo de pájaros, insectos o murciélagos.

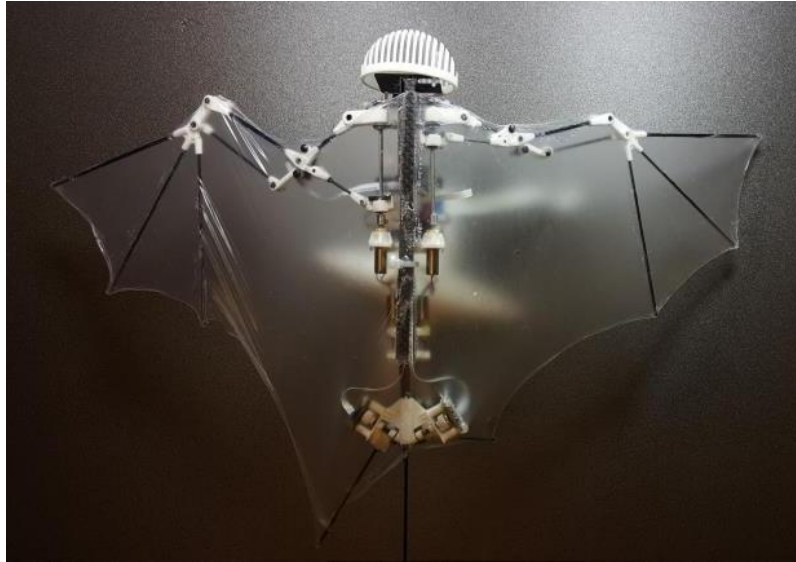


Fig. 1.2 Ejemplo dron de alas de aleteo con forma de murciélago [3]

- **Blimps o dirigibles:** estos UAVs no presentan una forma rígida. Su capacidad de vuelo se basa en la presión de los gases que contienen en el interior, No alcanzan altas velocidades, pero presentan mucha resistencia.

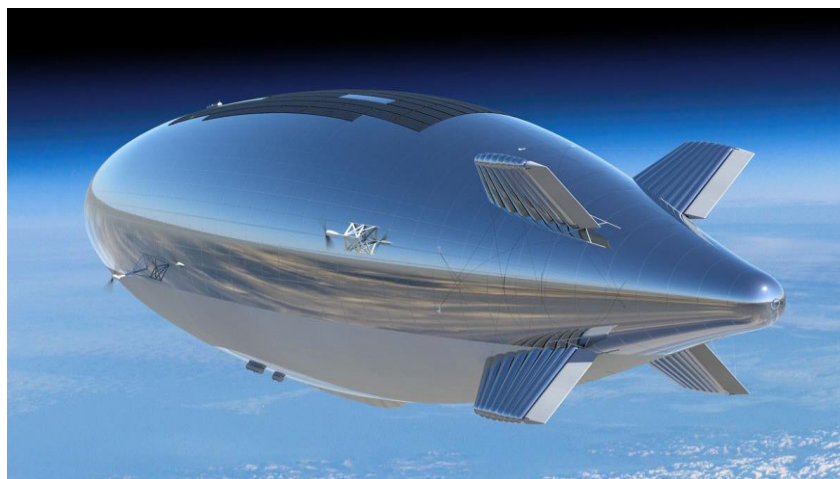


Fig. 1.3 Ejemplo dron dirigible [4]

- **Rotary-wings:** Son drones que presentan hélices que les permiten volar. Existen de diferente número de hélices: una (helicópteros), tres (tricopter), cuatro (quadrotor), seis (hexacopter) y ocho (octacopter) cuyo movimiento vertical viene definido por los movimientos individuales y contrapuestos de sus hélices como se aprecia en la siguiente imagen. Presentan alta maniobrabilidad y pueden aterrizar y despegar en vertical. [5, 6]



Fig. 1.4 Ejemplo dron hexacopter [7]

Existen varias configuraciones dentro de un mismo tipo de multicopter:

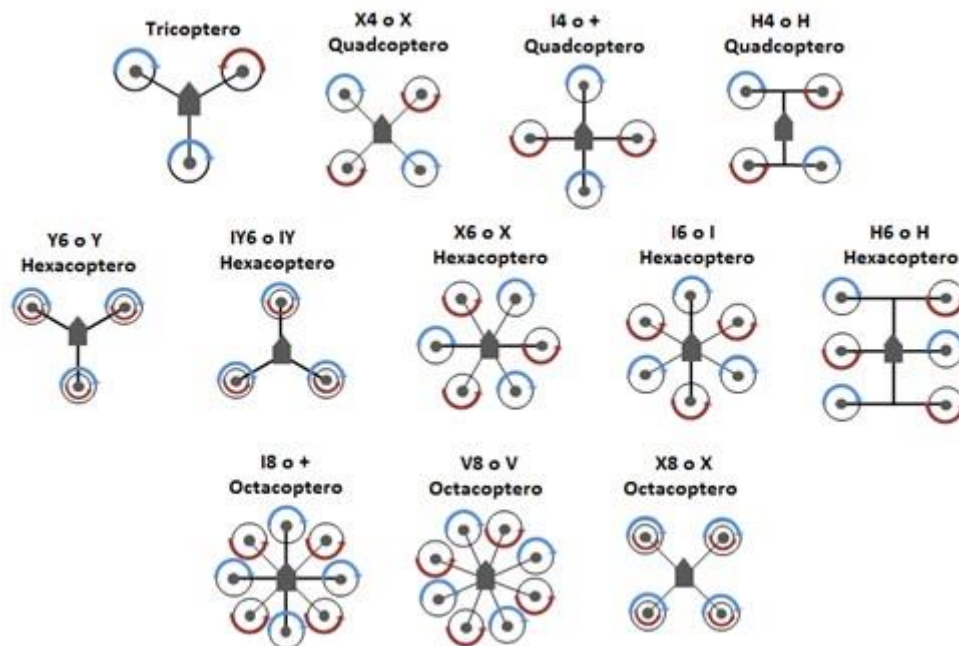


Fig. 1.5 Posibles combinaciones de las hélices [8]

En este proyecto se trabajará con un hexacopter en X (el del medio de la figura 1.5). En concreto, se realizará un aterrizaje autónomo en una plataforma que puede no ser completamente horizontal.

## **1.1 Motivación**

En los últimos años ha aumentado potencialmente el uso de este tipo de aeronaves, su creciente uso y su gran versatilidad han hecho de los drones una de las imágenes que constituyen nuestra visión del futuro, un cambio generacional desde el punto de vista tecnológico que se lleva fraguando desde hace años pero que resulta cada vez más tangible y más cercano.

Personalmente siempre me ha gustado la tecnología y en particular los drones; contribuir en este apasionante mundo representa para mi trabajar en lo que está siendo el futuro en la actualidad.

## **1.2 Problema**

El objetivo principal de este proyecto, como se ha mencionado anteriormente, es lograr que un Dron Hexacopter con configuración en X aterrice en una plataforma que pueda tener cierta inclinación.

En el problema que se aborda se pueden apreciar dos partes: la primera en la que el Dron recoge la información y la segunda, esa información se procesa, visualización y control respectivamente. Este proyecto se centra especialmente en la parte de control, aunque también se explicarán partes del proceso de visualización puesto que también forma parte del sistema de aterrizaje.

El Hexacopter debe partir de una posición dada, a continuación, debería reconocer un patrón de aterrizaje ('April tag') que le informe de la distancia a la que se encuentra el punto objetivo y a continuación deberá acercarse gradualmente cambiando su velocidad y ángulo de aproximación si fuera necesario, es decir, si el punto objetivo realmente estuviera inclinado. El proceso será satisfactorio una vez que el dron haya aterrizado sobre la plataforma.

## **1.3 Solución que se propone, Fuzzy Logic Controller (FLC) o controlador de lógica difusa:**

Para solucionar el problema planteado se propone un controlador basado en lógica difusa por su versatilidad, facilidad de configuración y buenos resultados. Además se requiere también del uso de

una placa Pixhawk la cual debe controlar la velocidad del dron y su ángulo de aproximación, una cámara que determine la posición del objetivo, ROS que es un framework para el desarrollo de *software* para robots que se distribuye en paquetes de los cuales vamos a usar MAVROS, el paquete Dron que es el controlador diseñado y en ausencia de un paquete que sea capaz de determinar la inclinación de la superficie se usará el paquete Ar\_track\_alvar para ejemplificar el controlador completo y funcional, todos estos componentes se explicarán más adelante [9].



## CAPÍTULO 2. ESTADO DEL ARTE

Una de las partes más importantes del proyecto ha sido la búsqueda y selección de métodos de control, ya que es una de las principales áreas de investigación en el campo de los vehículos no tripulados. Concretamente en el área de los hexacopter se ha investigado mucho en el tema del aterrizaje autónomo debido a la poca duración de su batería.

Los controladores siguen distintos algoritmos o ecuaciones para asegurar que se tiene un vuelo estable y de que se llega al punto deseado, pudiendo diferenciar controladores de alto nivel en los que los parámetros de salida son la velocidad o la posición final del dron y de bajo nivel donde los parámetros de salida son la velocidad de los motores, en este proyecto se va a trabajar con un control de alto nivel por su sencillez. Para conseguir los datos de entrada necesarios existen varias herramientas como pueden ser sensores internos tales como GPS, acelerómetros, osciloscopios... También existen dispositivos que pueden facilitarnos información crucial para efectuar un correcto procesamiento de la información exterior, aquí encontramos sensores ultrasonidos o cámaras de detección. [5, 6]

A continuación, se explican algunos de los controladores que se han estudiado y finalmente el controlador elegido:

### 2.1 Controlador clásico PID

El primer tipo de controlador que se pretende analizar es el PID por ser el más básico y común de todos. Se compone de tres controladores que, aparte de darle nombre, tienen diferentes características y funciones de transferencia, a saber:

1. Proporcional: la salida del controlador proporcional se define como el producto entre la ganancia que se quiera dar y la diferencia entre la posición deseada menos la actual, siguiendo la siguiente ecuación:

$$c(t) = K_e(t) \tag{2.1}$$

Donde  $c(t)$  es la salida,  $K$  es la ganancia proporcional y  $e(t)$  es el error en régimen permanente definido como  $x_d(t) - x_a(t)$ , siendo  $x_d(t)$  la situación final deseada y  $x_a(t)$  el estado actual del sistema.

2. Integral: el controlador integral toma el error integral de los valores de entrada y puede describirse como:

$$c(t) = \frac{1}{\tau_i} \int_0^t e(t) dt \quad (2.2)$$

Que usando la transformada de Laplace puede representarse como:

$$c(s) = \frac{1}{\tau_i s} e(s) \quad (2.3)$$

Donde  $\tau_i$  es la ganancia integral.

3. Diferencial: el controlador diferencial o derivativo toma la derivada del error anticipándose al mismo. Se define según la siguiente ecuación:

$$c(t) = \tau_d \frac{de(t)}{dt} \quad (2.4)$$

$$c(s) = \tau_d s e(s) \quad (2.5)$$

Donde  $\tau_d$  es la ganancia diferencial. La ganancia diferencial solo coge la derivada del error y afecta al estado futuro, pero no puede eliminar el error constante teóricamente. Por ello, no suele ser usado como sistema de control como tal, al contrario que el P que es el controlador más simple de las combinaciones posibles, suele dar sistemas estables y su funcionamiento radica en que aumentando la ganancia proporcional disminuye el error en régimen permanente,

sin embargo, un aumento excesivo puede provocar como resultado un sistema inestable debido a la sobre oscilación.

A veces, es posible crear un controlador basado únicamente en el integral. La ventaja de este controlador elimina es que el error, pero puede propiciar una respuesta muy lenta del sistema.

Lo habitual es que se combinen los tres tipos de controladores explicados previamente para suplir las posibles carencias que se puedan tener.

4. PI: una mezcla entre el proporcional y el integral que sigue la siguiente función:

$$c(t) = Ke(t) + \frac{K}{\tau_i} \int_0^t e(t)dt \quad (2.6)$$

y cuya función de transferencia es:

$$\frac{C(s)}{E(s)} = K \left( 1 + \frac{1}{\tau_i s} \right) \quad (2.7)$$

El controlador PI fusiona los dos efectos y hace que el error de posición sea nulo. La acción integral añade cero al lugar de las raíces y elimina el error de posición. Sin embargo, el efecto integral también afecta a la estabilidad del sistema en general así que se intenta que el efecto proporcional supla los efectos indeseados de la acción integral.

5. PD: o proporcional-derivativo cuya función de transferencia se rige por la siguiente ecuación:

$$c(t) = Ke(t) + K\tau_d \left( \frac{de(t)}{dt} \right) \quad (2.8)$$

y cuya función de transferencia es:

$$\frac{C(s)}{E(s)} = K(1 + \tau_d s) \quad (2.9)$$

En el caso del controlador PD el sistema actúa rápidamente debido a la acción derivativa corrigiendo el futuro error, sin embargo, el error de posición permanece.

6. PID: El controlador PID que combina las cualidades de los tres primeros, crea un sistema de respuesta rápido y elimina el error de posición siempre y cuando se hayan elegido las ganancias adecuadamente. El controlador PID responde ante la siguiente ecuación y tiene por tanto la siguiente función de transferencia [10]:

$$c(t) = Ke(t) + \frac{K}{\tau_i} \int_0^t e(t)dt + K\tau_d \frac{de(t)}{dt} \quad (2.10)$$

$$\frac{C(s)}{E(s)} = K\left(\frac{1}{\tau_i s} + 1 + \tau_d s\right) \quad (2.11)$$

## 2.2 Controlador LQR

El controlador *Linear Quadratic Regulator* o LQR por sus siglas, consiste en un criterio de minimización usando una retroalimentación que se rige por la función:

$$u = -kx(t) \quad (2.12)$$

El índice de rendimiento que se utiliza para minimizar tanto el esfuerzo de control como los estados se definen:

$$J = \frac{1}{2} \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (2.13)$$

$\|x\| = x^T(t)x(t)$  y  $x$  son matrices  $n \times 1$  que se asocian al vector estado. También se incluyen  $u$  entradas para controlar el sistema. Las matrices  $Q$  y  $R$  son llamadas matrices de peso y nos permiten definir el efecto de cada estado y la entrada de control en la respuesta del controlador cuando usamos la retroalimentación para diseñarlo.

$Q$  es una matriz  $n \times n$  real, simétrica y positiva o semipositiva:

$$Q = \begin{bmatrix} q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & q_n \end{bmatrix} \quad (2.14)$$

El peso dado por el elemento  $i$ -ésimo es una medida del esfuerzo de control para controlar el estado relacionado, cuanto más grande el elemento más grande el esfuerzo de control usado para el estado.

La matriz  $R$  es una matriz  $p \times p$  real, simétrica, positiva y definida.

$$R = \begin{bmatrix} r_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & r_p \end{bmatrix} \quad (2.15)$$

Si es posible encontrar una función de retroalimentación adecuada, que como se ha definido previamente sigue la ecuación  $u = -kx(t)$ , que minimice  $J$  para el sistema dado por  $\dot{x} = Ax + Bu$  entonces sería posible encontrar una solución óptima al problema de control, para ello sería necesario seguir el criterio de estabilidad de Lyapunov.

Sin entrar en demasiado detalle, el criterio de estabilidad de Lyapunov determina que si la respuesta de un sistema no vibratorio desde un punto  $x$  va cerca del punto  $e^x$  para todo  $x$  entonces el sistema es estable desde el punto de vista del criterio de Lyapunov [11].

## 2.3 Controlador basado en lógica difusa

Mientras que la lógica tradicional está basada en valores que son 0 o 1, verdadero o falso, sí o no, donde los límites de los conjuntos están claramente definidos, la lógica difusa es una lógica multivariable y está basada en conceptos de conjuntos difusos con límites vagos cuyos elementos pueden oscilar entre 0 y 1. Esta característica proporciona al sistema la posibilidad de establecer un razonamiento mucho más parecido al razonamiento humano del que tenía la lógica clásica ya que está basado en verdades parciales. Como ejemplo, mientras que en la lógica clásica el agua está o fría o caliente en la lógica difusa el agua puede estar templada incluso pudiendo categorizarse ese nivel de templado, ya que 0,3 sería templado tirando a frío y 0,7 sería un templado que se acercaría al caliente.

Un controlador basado en lógica difusa presenta las siguientes partes:

- Variables lingüísticas: son variables cuyos valores son palabras o frases en un lenguaje natural o artificial. Aplicando al concepto de temperatura puede tener valores como caliente, frío, templado, templado-frío, templado-caliente; otro ejemplo que puede aplicarse a la señal de salida es el de la potencia, puede obtener valores como mínimo, máximo, medio o incluso valores relativos más bajo, más alto...
- Conjunto difuso: el conjunto difuso  $A$  en el universo  $X$  se define como un conjunto de pares ordenados:

$$A = \{x, \mu_A(x); x \in X\} \quad (2.15)$$

donde  $\mu_A$  es llamado función de membresía para el conjunto difuso  $A$ , la cual determina el grado de pertenencia de  $x$  al conjunto  $A$ .

$$\mu_A(x): X \rightarrow [0, 1] \quad (2.16)$$

- Funciones de membresía: definen los rangos de las variables lingüísticas y sus valores. Para ello, se utilizan representaciones geométricas como funciones de Gauss, funciones triangulares, trapezoidales, en rampa, etc. En la siguiente imagen se pueden apreciar las diferencias entre una función de membresía de la lógica clásica que, como se ha mencionado anteriormente, tiene los límites claramente definidos y la función de membresía que se puede encontrar en la lógica difusa siguiendo en este ejemplo una distribución gaussiana.

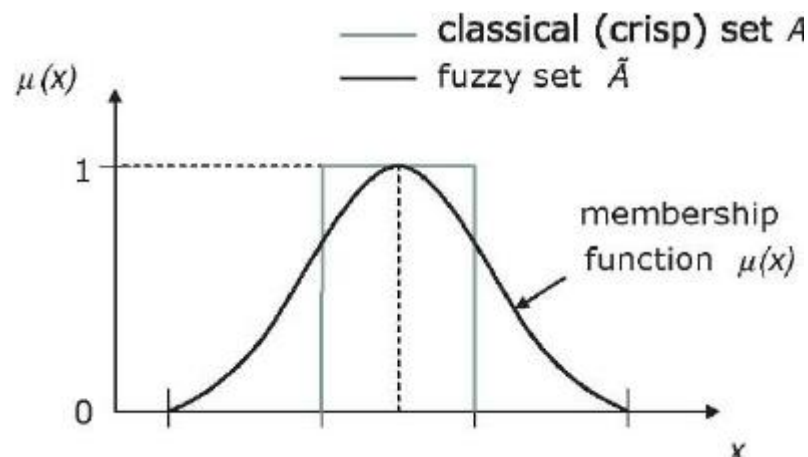


Fig. 2.1 Gráfico comparativo entre conjunto difuso y conjunto lógico clásico [12]

- Reglas: definen la relación entre las variables lingüísticas de entrada y salida. Un sistema difuso se compone de un conjunto de reglas que relacionan premisas con consecuencias. Las reglas siguen la siguiente estructura: SI “premisas” ENTONCES “consecuencias”. Las premisas representan comparaciones entre una variable lingüística de entrada y valores de sus funciones de membresía. Para convertir un valor numérico de entrada en variable lingüística se da un proceso que llamado ‘*fuzzification*’, acto seguido se aplican las consecuencias. Una vez que se han evaluado todas las reglas, las consecuencias son evaluadas de forma conjunta. El valor cuantitativo de la salida se produce con otro proceso llamado ‘*defuzzification*’ y ese valor numérico obtenido se puede aplicar al proceso en cuestión.

Por tanto y como se ha mencionado con anterioridad, la ejecución de un sistema difuso comprende los siguientes tres procesos:

- *Fuzzification*: es el proceso en el cual un grupo de valores obtenidos de las señales de entrada al sistema son comparados con los valores de cada función de membresía para determinar el nivel de correspondencia que pueda tener con la variable lingüística en cuestión. Es decir, mediante la función de membresía, se asocia el valor numérico de entrada a una variable lingüística. En el ejemplo de la temperatura, se asocia 30° a caliente.
  
- Evaluación de reglas: es el proceso por el cual se analizan todas las reglas del sistema difuso. Ello conlleva que la variable lingüística de entrada active un valor en cada premisa calculando el valor de salida o consecuencia para cada regla. La activación parcial de premisas resulta en la activación parcial de consecuencias que da como resultado un área que surge de la combinación de las consecuencias activadas parcialmente. Existen tres tipos de modelado para reglas difusas:
  - ❖ Tipo Mandami: Este tipo de reglas incluye conocimiento semicualificado sobre las variables y se definen de la siguiente forma:

$$\text{SI } x \text{ A}_i \text{ ENTONCES } y \text{ B}_i \quad (2.17)$$

En esta regla,  $x$  es la entrada lingüística y  $A_i$  son los antecedentes en términos lingüísticos. Del mismo modo  $y$  es una salida lingüística y  $B_i$  son las consecuencias en términos lingüísticos. Las variables lingüísticas  $x$  e  $y$  y los términos lingüísticos  $A_i$  y  $B_i$  en los conjuntos difusos son definidos con las funciones de membresía. Por ejemplo:

Teniendo los conjuntos  $A = \{\text{calor, templado, frío}\}$  y  $B = \{\text{subir temperatura, bajar temperatura, no cambiar temperatura}\}$  de un regulador térmico doméstico las reglas siguiendo un modelo Mandami serían:

Regla 1: SI temperatura ( $x$ ) es calor ( $A_1$ ) ENTONCES sistema ( $y$ ) bajar temperatura ( $B_2$ )

Regla 2: SI temperatura ( $x$ ) es templado ( $A_2$ ) ENTONCES sistema ( $y$ ) no cambiar temperatura ( $B_3$ )



Regla 3: SI temperatura (x) es frío ( $A_3$ ) ENTONCES sistema (y) subir temperatura ( $B_1$ )

- ❖ Tipo Takagi-Sugeno: En las reglas de tipo Mamdani los antecedentes y las consecuencias definen las proposiciones difusas. En las consecuencias en las reglas de tipo Takagi-Sugeno se tiene como salida una función.

Su estructura es:

$$\text{SI } x \text{ es } A_i \text{ ENTONCES } y_i = f_i(x) \quad i = 1, 2, 3 \dots, r \quad (2.18)$$

Todas las funciones  $f(x)$  deben ser del mismo tipo, sin embargo, los parámetros pueden variar entre ellas. Es común usar funciones lineales por simplicidad.

- ❖ Tipo Tsukamoto: Cuando se modela bajo las premisas de las reglas del tipo Tsukamoto encontramos que si de base se partía como en el modelo Mamdani con la siguiente estructura:

$$\text{SI } x \text{ es } A_i \text{ ENTONCES } y \text{ es } B_i \quad i = 1, 2, 3 \dots, r \quad (2.19)$$

Puede ser mostrado como una relación  $R_i: (X \times Y) \rightarrow [1, 0]$

Esta relación puede ser calculada usando relaciones de intersección. Si se utiliza el operador mínimo para la intersección, la función de membresía para  $R_i = A_i \times B_i$  puede ser definida usando las funciones de membresía de  $A$  y  $B$ , así pues, se tendría que:

$$\mu_{R_i}(x, y) = \mu_{A_i}(x) \wedge \mu_{B_i}(y) = \min(\mu_{A_i}(x), \mu_{B_i}(y)) \quad (2.20)$$

El operador mínimo es aplicado para todas las posibles combinaciones  $x - y$  en el espacio cartesiano de  $X$  e  $Y$ . Las relaciones  $R$  que definen el modelo son dadas por la unificación de  $r$  reglas difusas en base a

$$R = \bigcup_{i=1}^r K_i \quad (2.21)$$

Si se utiliza el operador máximo para la unificación la función de membresía para la relación de  $R$  viene dada por:

$$\mu_R(x, y) = \max_{1 \leq i \leq r} (\mu_{A_i}(x) \wedge \mu_{B_i}(y)) = \max_{1 \leq i \leq r} (\min(\mu_{A_i}(x), \mu_{B_i}(y))) \quad (2.22)$$

Es posible usar composiciones min-máx. para obtener la eficacia del factor de salida. En este caso se elige el valor más pequeño de los valores de entrada y se calcula el área de la función de membresía. La utilización de más de una regla es mucho más efectiva porque cada valor de salida puede ser usado para calcular una media ponderada que dé el resultado final.

- *Defuzzification*: es el proceso que se encarga de obtener el valor numérico de salida. Para calcular dicho valor primero y como se ha comentado previamente, se crea un área con la unión de cada función de membresía de salida usada. Existen varios métodos para calcular el valor de salida. El usado más habitualmente y el que se va a implementar en este proyecto es el cálculo del centro de gravedad del área. Dependiendo del problema puede interesar usar el centro geométrico, el máximo más largo, el máximo más corto o la mitad del máximo. Todos estos métodos para calcular el valor de salida en base a un área obtenida son mostrados en la siguiente ilustración [13, 14]:

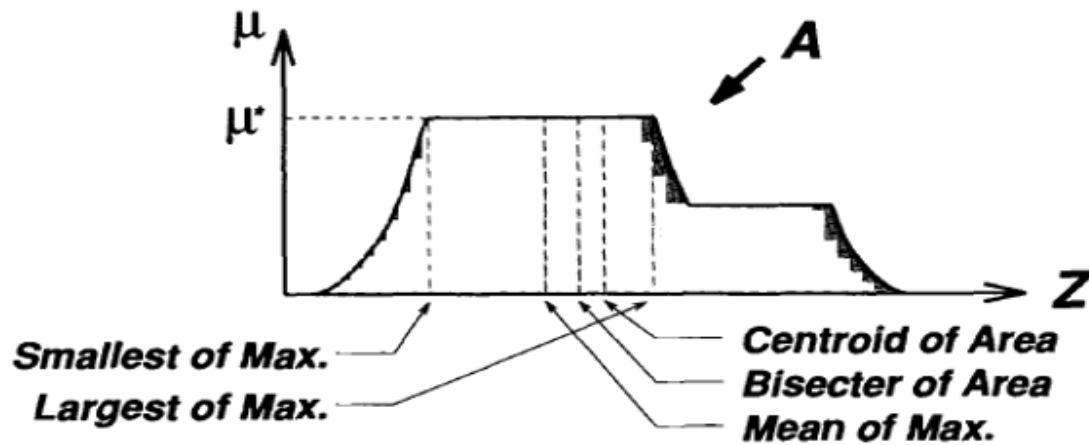


Fig. 2.2 Comparativa entre los distintos métodos del proceso de 'Defuzzification' [15]

La figura 2.3 presenta de forma esquemática cómo funciona un sistema basado en lógica difusa:

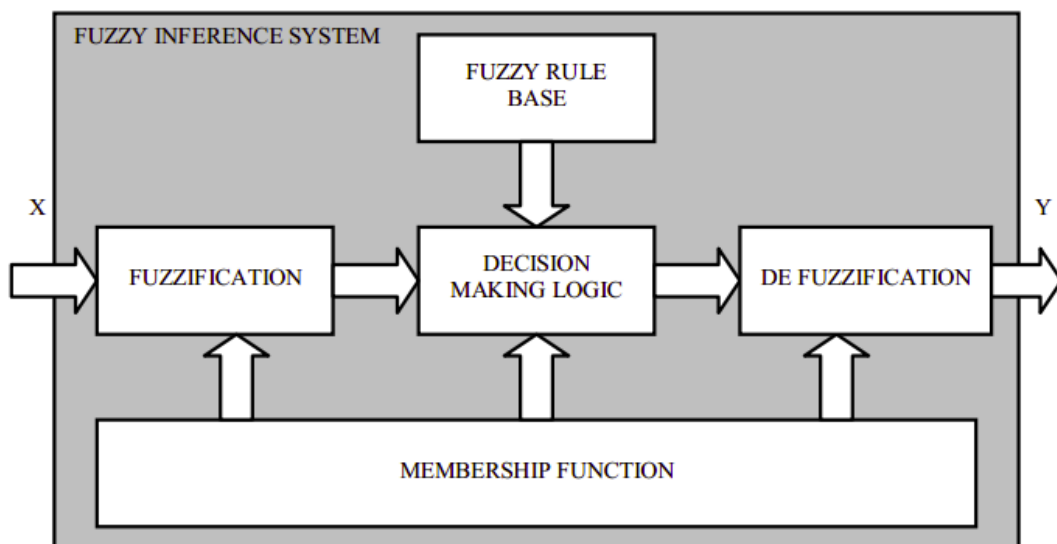


Fig. 2.3 Esquema genérico de un sistema basado en lógica difusa [16]

Como se ha señalado, en este proyecto se utiliza el método de lógica difusa. Se ha elegido este método de control por su versatilidad y facilidad de implementación al problema propuesto. Además, un sistema basado en lógica difusa permite ajustar las funciones de membresía sin necesidad de conocer los parámetros del problema lo cual en este caso donde no se tiene un sistema de visualización adecuado, hacen de este controlador una solución ideal.

## CAPÍTULO 3. ALGORITMO PROPUESTO

En este capítulo primero se describen las ecuaciones utilizadas posteriormente se describe el proceso para la creación de un controlador basado en lógica difusa, luego el algoritmo de control propuesto y por último la explicación del programa.

El controlador ha sido programado en C++ e implementado en ROS un entorno de trabajo que se explicará más adelante. Se han utilizado también las librerías y el programa de lógica difusa del proyecto [14] cuyo programa se explicará también en este apartado.

### 3.1 Ecuaciones

En esta sección se analizan las ecuaciones empleadas para el correcto aterrizaje del dron. El problema planteado es un típico problema de rotación de sistemas de referencia, así pues, mientras la superficie objetivo no tenga ningún tipo de inclinación el sistema de coordenadas del dron y el sistema de referencia inercial que se ha situado en el centro del dron, con la orientación que determina la Pixhawk, será el mismo. Sin embargo, si se diera el caso de que la plataforma de aterrizaje estuviera inclinada sería necesario rotar el sistema de referencia del dron alterando su vector velocidad respecto al primer sistema de referencia propuesto, planteando una trayectoria diagonal.

Si se tuviera un correcto sistema de visualización, no importaría que el dron rotase, las coordenadas visualizadas vendrían dadas por en el sistema de referencia de la aeronave y la velocidad de salida consecuentemente, también. Sin embargo, en ausencia de dicho sistema se ha tenido que calcular la nueva posición en función del avance, como se muestra en la figura 3.4 de este apartado, y por tanto, se requiere de matrices de cambio de base para la realización del problema.

Conocida la distancia inicial, se requiere cambiar dichas coordenadas al sistema de rotación del dron para que el controlador de lógica difusa pueda calcular unas velocidades de salida que serán las que se apliquen al dron. No obstante, esas velocidades y desplazamientos son complicadas de evaluar desde el punto de vista de un observador que no rota como lo hace el vehículo. Es por lo que se ha decidido

una vez que se tienen las velocidades  $V_x'$ ,  $V_y'$  y  $V_z'$  del dron volver a transformarlas al sistema de referencia inicial, calculando el avance en este sistema de referencia y entrando en el bucle cerrado que se muestra en la figura 3.4.

En primer lugar, se tiene en cuenta que en la plataforma no existe rotación en el eje Z ya que al ser paralelo al plano horizontal no alteraría la horizontalidad de la plataforma. En segundo lugar, se multiplican las matrices de cambio de base para cuando existe rotación sólo en uno de los ejes, primero en X y luego en Y,  $\alpha$  y  $\beta$  respectivamente.

Cabe destacar que el sistema de referencia que ofrece la Pixhawk tiene un eje X que entra en el papel por lo que al plantear la rotación  $\alpha$  y  $W_x$  tendrían sentidos contrarios a los que acostumbra el lector, siendo horario el de signo positivo y negativo el de sentido antihorario.

Con el fin de ilustrar mejor las rotaciones se han añadido imágenes en dos dimensiones de los dos escenarios, el primero muestra el plano ZY y el segundo muestra el plano ZX. En ambos, en negro, se representa el sistema de referencia sobre el que se va a trabajar y, en azul, el sistema de referencia del dron una vez ha alcanzado la inclinación adecuada para el aterrizaje cuando se da dicha inclinación en la plataforma la cual es representada en forma de triángulo en la esquina superior derecha del dibujo.

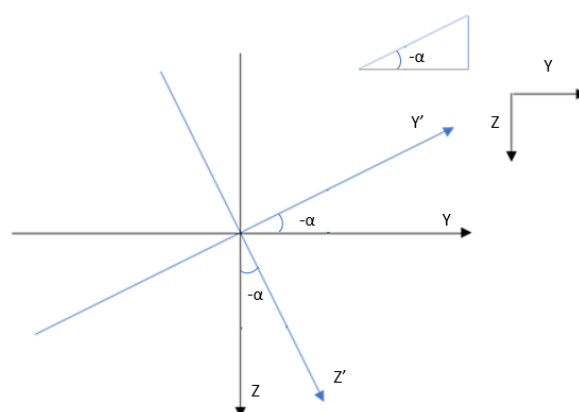


Fig. 3.1 Plano ZY

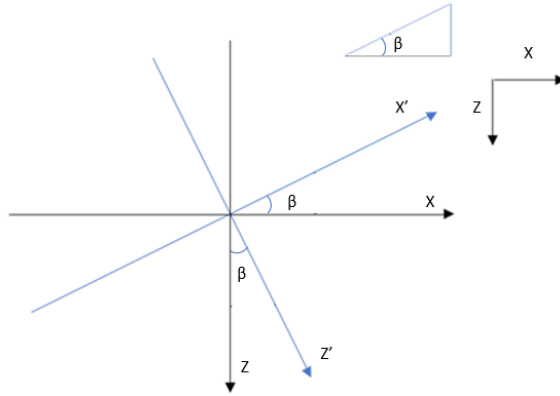


Fig. 3.2 Plano ZX

Las matrices de cambio de base cuando existen rotaciones en el eje X y en el Y son:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\operatorname{sen} \alpha \\ 0 & \operatorname{sen} \alpha & \cos \alpha \end{bmatrix} \quad (3.1)$$

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \operatorname{sen} \beta \\ 0 & 1 & 0 \\ -\operatorname{sen} \beta & 0 & \cos \beta \end{bmatrix} \quad (3.2)$$

Si se multiplica  $R_x(\alpha)R_y(\beta)$  se obtiene la matriz de cambio de base definida como:

$$R_x(\alpha)R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \operatorname{sen} \beta \\ \operatorname{sen} \beta \operatorname{sen} \alpha & \cos \alpha & -\operatorname{sen} \alpha \cos \beta \\ -\operatorname{sen} \beta \cos \alpha & \operatorname{sen} \alpha & \cos \alpha \operatorname{sen} \beta \end{bmatrix} \quad (3.3)$$

Con lo que se obtendría la siguiente ecuación:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \beta & 0 & \operatorname{sen} \beta \\ \operatorname{sen} \beta \operatorname{sen} \alpha & \cos \alpha & -\operatorname{sen} \alpha \cos \beta \\ -\operatorname{sen} \beta \cos \alpha & \operatorname{sen} \alpha & \cos \alpha \operatorname{sen} \beta \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3.4)$$

Lo que daría el siguiente sistema de ecuaciones:

$$x' = x \cos \beta + z \sin \beta \quad (3.5)$$

$$y' = x \sin \beta \sin \alpha + y \cos \alpha - z \sin \alpha \cos \beta \quad (3.6)$$

$$z' = -x \sin \beta \cos \alpha + y \sin \alpha + z \cos \alpha \cos \beta \quad (3.7)$$

Teniendo en cuenta que los ángulos se pueden definir como la velocidad angular por la variación de tiempo se obtienen las ecuaciones definitivas que van a utilizarse en este proyecto:

$$x' = x \cos(w_y \Delta t) + z \sin(w_y \Delta t) \quad (3.8)$$

$$y' = x \sin(w_x \Delta t) \sin(w_y \Delta t) + y \cos(w_x \Delta t) - z \sin(w_x \Delta t) \cos(w_y \Delta t) \quad (3.9)$$

$$z' = -x \sin(w_y \Delta t) \cos(w_x \Delta t) + y \cos(w_x \Delta t) \cos(w_y \Delta t) \quad (3.10)$$

Nótese que para devolver el vector posición y el vector velocidad al sistema de referencia inicial basta con invertir el signo de los ángulos.

### 3.2 Modo de trabajo para diseñar un controlador

El diseño del controlador de lógica difusa se basa en un proceso de trabajo circular que sigue el esquema que se adjunta a continuación. En primer lugar, se modifican las funciones de membresía de los parámetros y/o las reglas. En segundo y tercer lugar, la respuesta del sistema es simulada y evaluada. Este proceso se repite hasta que se encuentra un controlador aceptable. Una vez que las simulaciones sean satisfactorias el último paso consiste en probarlo en el *hardware* y evaluar su respuesta modificando los parámetros pertinentes hasta encontrar el controlador óptimo [9].

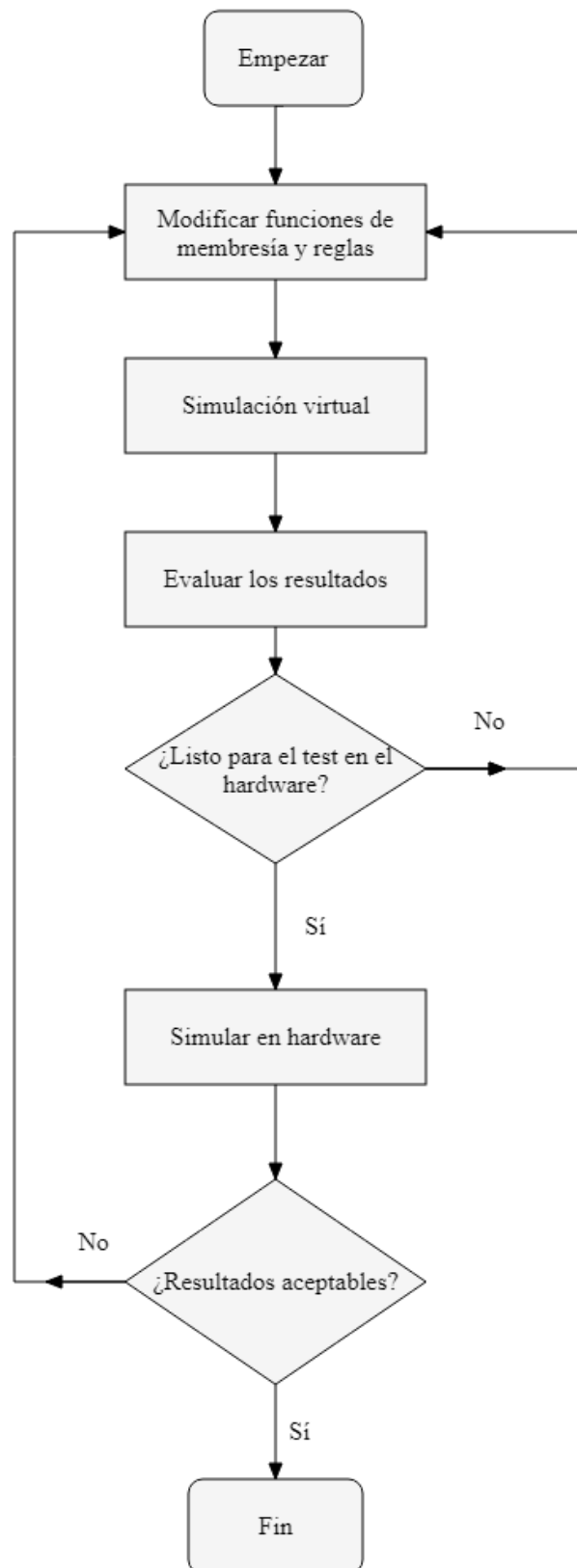


Fig. 3.3 Workflow para diseñar un sistema de lógica difusa [9].



### 3.3 Algoritmo del controlador

En ausencia del sistema de visualización necesario para determinar la posición relativa del dron respecto a la superficie objetivo se ha propuesto un sistema de control en bucle cerrado donde el *feedback* viene definido por la posición final o posición objetivo, que es conocida desde el principio y es invariable, y la posición actual que se define gracias a la distancia que avanza el dron debido a una velocidad de salida. Esto permite obtener la distancia entre el dron y la plataforma final creando una velocidad de movimiento acorde a dichos resultados [9]. Como se ha comentado previamente, dentro del controlador difuso se cambian las coordenadas al sistema de referencia del dron y posteriormente se devuelven las velocidades al sistema inicial para evaluar su avance.

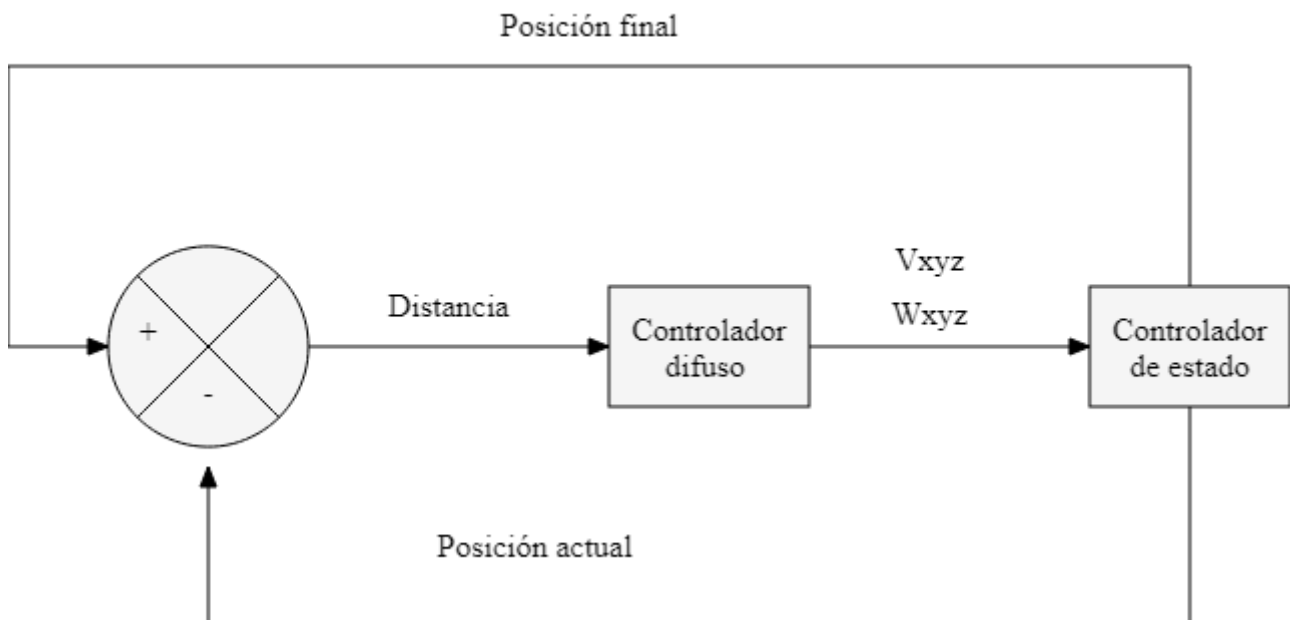


Fig. 3.4 Representación del sistema de control de lógica difusa [9].

A continuación, se adjunta un *flowchart* que mostraría la arquitectura ROS del programa esto es el flujo de trabajo con los *topics* y en ausencia de un paquete de visualización apto, se pondrá de ejemplo el paquete `Ar_track_alvar` por ser el que más se ajusta a las necesidades del proyecto. En el diagrama se pueden apreciar los *topics* que utilizan los paquetes siendo `/visualization_maker/pose/position` el que pasaría los datos de entrada del paquete `Ar_track_alvar` al paquete `Dron` creado y siendo `/setpoint_velocity/cmd_vel_unstamped/geometry_msgs/Twist` el que pasaría los datos respectivos a la velocidad del paquete creado al paquete `MAVROS` [6]. En la sección de 4.1.1 ROS de este proyecto se explican más detenidamente algunos términos usados en esta sección:

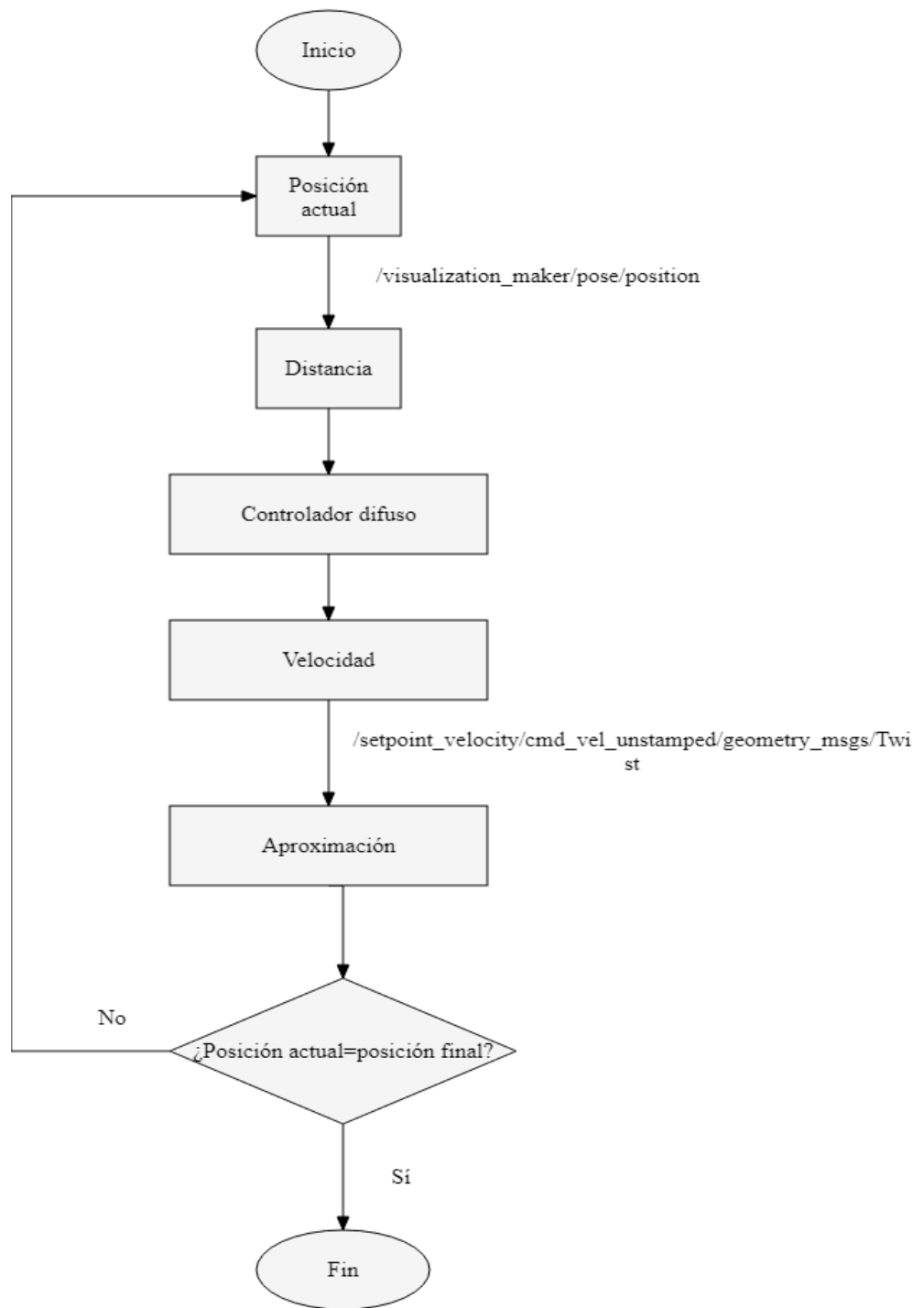


Fig. 3.5 Workflow del controlador de lógica difusa incluyendo topics [6].

### 3.4 Librerías lógica difusa

El programa de lógica difusa se desarrolló en el trabajo [14]. Es un paquete libre de ROS. Cabe destacar que lo más importante se encuentra en las carpetas *include*, donde se encuentran los archivos .h, es decir, las librerías, en la carpeta *src*, donde se encuentran los .cpp, que son los archivos que contienen el código fuente en C++, y en la carpeta *params* donde se encuentran archivos con extensión .fz que es donde se encuentran las reglas y las funciones de membresía tanto de entrada como de salida. Estas funciones están compuestas al principio por el nombre de la variable lingüística, posteriormente por el valor lingüístico y por último de cuatro números que definen la función de membresía, siendo los dos centrales iguales en caso de tratarse de funciones de membresía triangulares.

En lo que respecta al funcionamiento *software* se adjunta a continuación el diagrama UML

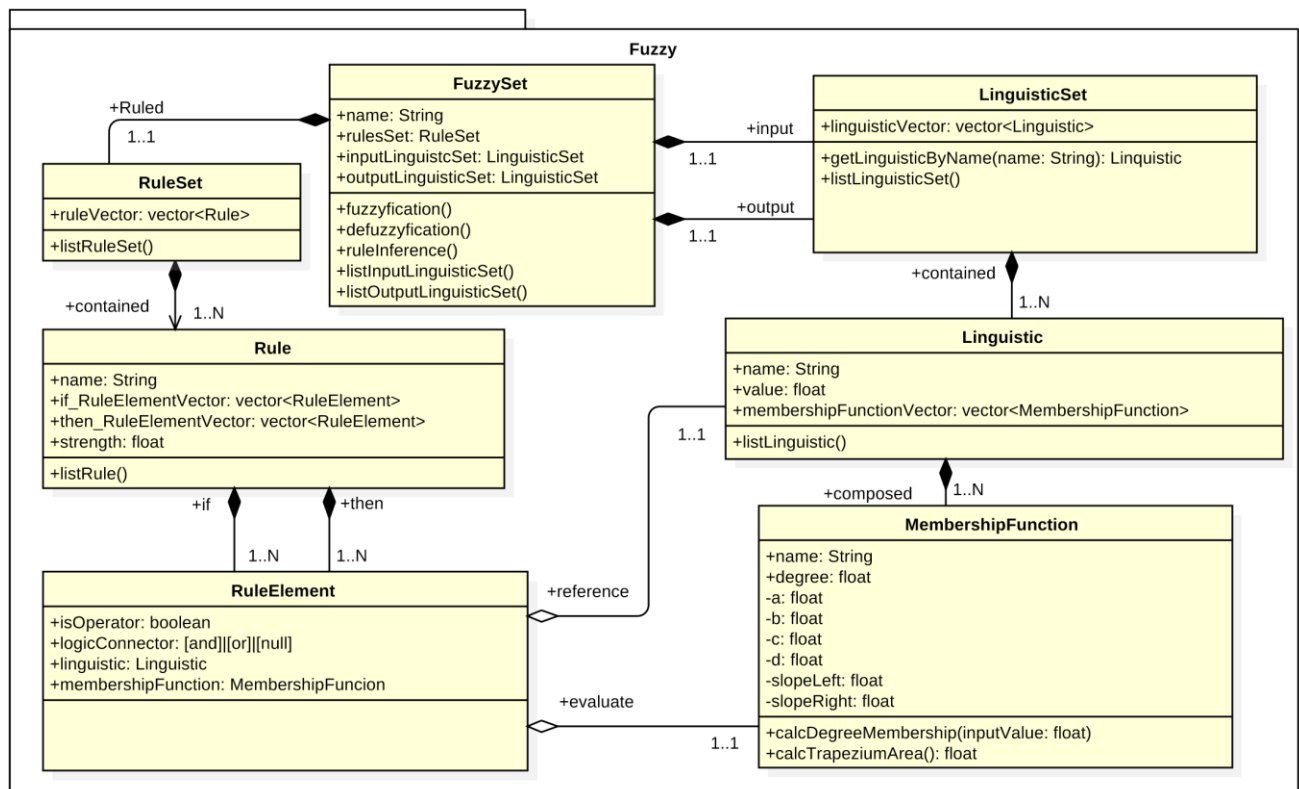


Fig. 3.6 Diagrama UML de las clases de las librerías de lógica difusa [14].

Para implementar el código al proyecto, se ha creado una clase de tipo *dron* que tiene 6 clases *fuzzyset* que utiliza la función `initFZ()` para cargar los parámetros y las reglas en cada conjunto difuso para las direcciones X, Y, Z y los ángulos de  $\alpha$ ,  $\beta$  y  $\gamma$  asociados a la rotación de los ejes de coordenadas respectivamente.

## CAPÍTULO 4. RESULTADOS EXPERIMENTALES

### 4.1 Plataformas

Para implementación de un controlador basado en lógica difusa de alto nivel se han utilizado diferentes plataformas *Hardware* y *Software* que se comentan a continuación:

#### 4.1.1 ROS

ROS o *Robot Operating System* por sus siglas en inglés, es un *framework* de código abierto. ROS provee todos los servicios que se podrían esperar de un sistema operativo incluyendo abstracción *hardware*, control de dispositivos de bajo nivel, implementación de funcionalidad de uso común, paso de mensajes entre procesos y mantenimiento de paquetes.

ROS está basado en una arquitectura de grafos donde los procesos se llaman nodos y la información que intercambian, mensajes. Los mensajes son de distintos tipos, desde mensajes de tipo entero (int) o cadenas de caracteres (string) hasta librerías más complejas como *geometry\_msgs* la cual está compuesta por *arrays* de mensajes más simples, tienen una sola dirección y se envían a través de lo que se denomina *topic*. Los *topics* básicamente son canales de información y, como se puede apreciar en la imagen adjunta, un nodo puede publicar o suscribirse a un *topic* dependiendo de si se requiere que envíe información o la reciba respectivamente.

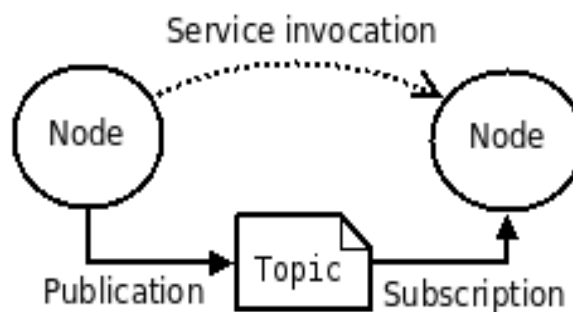


Fig. 4.1 Esquema del flujo de información de ROS [17].

En el controlador que se ha creado primero hay un nodo llamado `Ar_track_alvar` que se encarga de determinar la distancia del dron a la plataforma gracias a una cámara y al ‘April tag’. Dicha información se envía por un *topic* llamado `"/visualization_marker/pose/position/x"` en el caso de la coordenada x, al nodo principal *main* donde el programa de lógica difusa procesa la información y determina una velocidad de salida. Esta velocidad de salida se envía por el *topic* `"/setpoint_velocity/cmd_vel_unstamped/geometry_msgs/Twist"` (esta vez las seis velocidades de los tres ejes a un solo *topic*) y se suscribe al nodo MAVROS [17]. Como se ha señalado previamente, en el proyecto se utiliza el paquete `Ar_track_alvar` en defecto de uno que pueda determinar la inclinación de la plataforma, pero el modo de operación sería muy similar.

#### 4.1.2 *Ar\_track\_alvar*

Este paquete de ROS permitiría al dron reconocer la plataforma objetivo y calcular su distancia en los tres ejes, basándose en la detección del ‘April Tag’. Para ajustar la detección se puede optar por combinar múltiples ‘April Tag’, unos dentro de otros como se muestra en la figura 4.2. Con grandes distancias, entre 0,5 y 1,5 metros, la cámara captaría demasiado ruido como para detectar los ‘April Tag’ más pequeños y solo detectaría el grande. Para distancias menores a 0,5 metros el ‘April Tag’ más grande no sería reconocido por el paquete `Ar_track_alvar` pero los pequeños sí. Esto permitiría un ajuste mucho mejor y más preciso del controlador [9, 18].



Fig. 4.2 'April Tag' combinado [9].

### 4.1.3 MAVROS

Es un paquete de ROS que le permite interactuar con el *software* de piloto automático utilizando el protocolo MAVLink. Es decir, sirve como nexo entre el sistema operativo del robot y la controladora PX4 como se puede apreciar en la imagen 4.3:

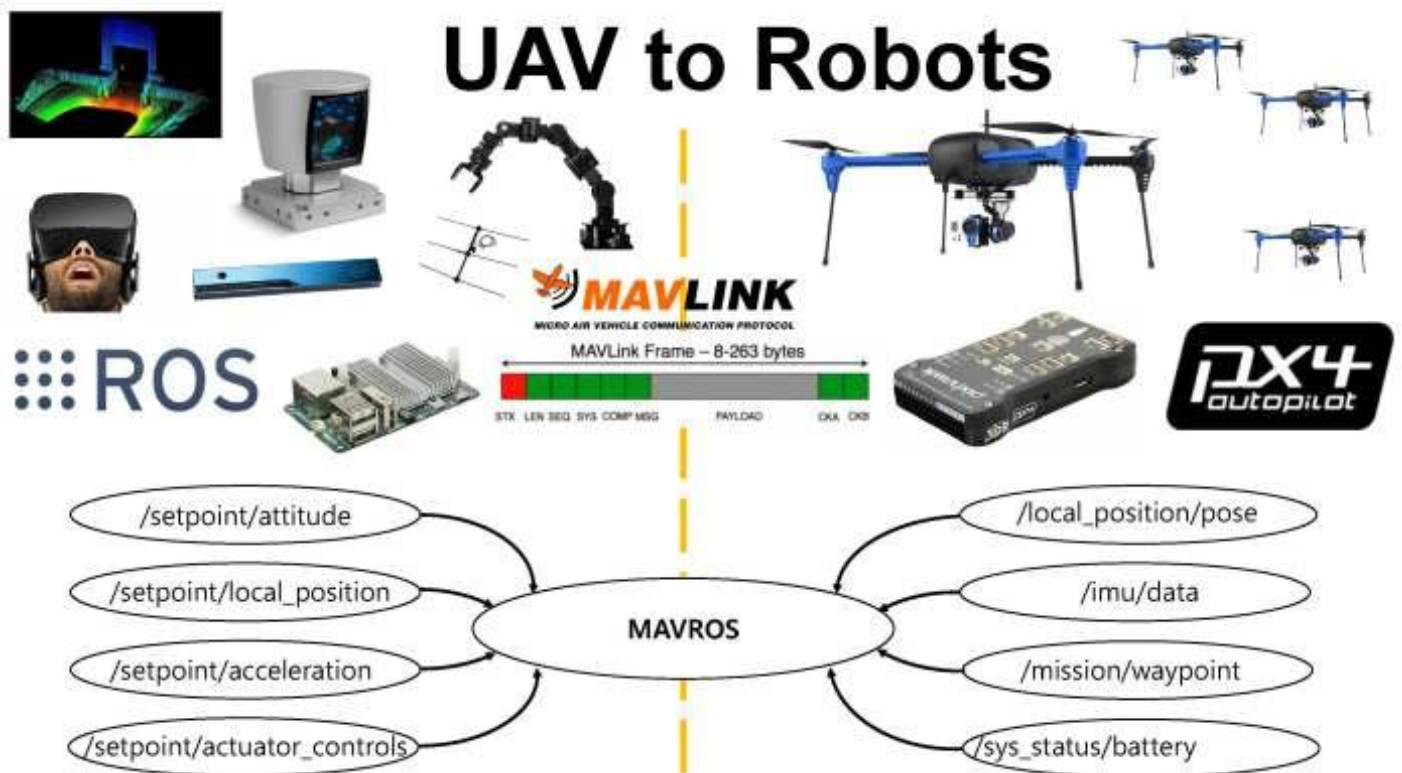


Fig. 4.3 Esquema de MAVROS [19]

El paquete de MAVROS incluye las siguientes características:

- Comunicación con el piloto automático mediante un puerto serial.
- *Proxy* UDP, para la estación de control de tierra.
- Tópicos compatibles con MAVLink\_ros

- Sistema *plugins* para ROS-MAVLink
- Herramienta de manipulación de parámetros
- Herramienta de manipulación de puntos de interés [6, 20].

#### 4.1.4 MAVLink

Es un protocolo de mensajería muy ligero para comunicarse con drones y ordenar librerías de los UAVs. Sigue un modelo de patrón híbrido de publicación y suscripción y punto a punto. Dicha información se encuentra contenida en ficheros de extensión .XML lo que permite que pueda ser estructurada y empaquetada empleando diferentes lenguajes de programación como C++, C, Python, Java, Lua, entre otros. Sirve como intermediario para la comunicación tanto de la IMU/MCU del dron como para la estación de tierra de comunicación en Linux.

El protocolo MAVLink mencionado en el apartado de MAVROS consiste en tres: cabecera, *payload* y *checksum*.

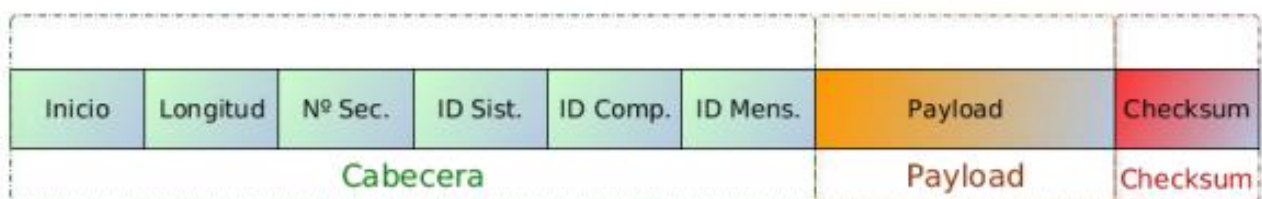


Fig. 4.4 Estructura de un mensaje típico que sigue el protocolo MAVLink [21].

La cabecera está compuesta por:

- Inicio: Indica el comienzo del paquete MAVLink
- Longitud: determina el número de bytes que se van a enviar como *payload* del mensaje. Toma valores entre 0 y 255.
- Numero de secuencia: Muestra el número de paquete enviado según su pertenencia a la secuencia de estos para reorganizarlos en el destino o para detectar paquetes perdidos. Tiene un rango de valores entre 0 y 255.
- Identificación del sistema: permite diferenciar varios sistemas de la misma red ya que identifica el sistema con un número natural. Su rango de valores es de 0 a 255
- Identificación del componente del sistema: señala mediante un número natural el componente que ha enviado el mensaje. Toma valores entre 0 y 255.
- Identificación del mensaje: identifica el tipo de mensaje enviado indicando como se va a leer el *payload* que viene a continuación.

En el *payload* es donde se encuentra la información que se quiere transmitir. El control de errores se realiza a partir de la información del *Checksum* [21, 22].

#### 4.1.5 PIXHAWK PX4

PX4 inicialmente fue diseñado para ejecutarse en controladores de la serie Pixhawk. Esta serie es un proyecto que crea diseños *hardware* abiertos que definen un conjunto de componentes (CPU, sensores, etc.) y sus conexiones, cuya finalidad es proporcionar controladores de piloto automático de alta disponibilidad y bajo coste, por lo que se utiliza en proyectos de investigación, industria y hobbies [23, 24].



Así pues, la PX4 es un potente código abierto que cumple funciones de vuelo automático. Algunas de sus características son:

- Controla muchos tipos diferentes de vehículos no tripulados tanto terrestres como aéreos como submarinos.
- Acepta una amplia variedad de *hardware* para controlar el estado del vehículo (posición, altitud, dirección, velocidad, nivel de batería, velocidad del aire, etc.). Para ello usa sensores implementados entre los que encontramos giróscopos, acelerómetros, magnetómetros, barómetros [25] y sensores periféricos como sensores GPS y compás, sensor de velocidad aerodinámica, sensores de distancia o cámaras para tratamiento de imagen [26].
- Controla los actuadores del dron en este caso los motores, que son accionados a través de un controlador electrónico de velocidad o ESC (*Electronic Speed Controller*). El ESC convierte una señal del controlador de vuelo a un nivel apropiado de potencia para el motor [24].
- Separa el uso de la batería entre las funciones del controlador de vuelo y los ESC para los motores [24].
- Aunque el PX4 no requiere de sistema de control remoto para vuelos autónomos la placa también permite controlar manualmente el vehículo mediante un sistema de radio control (RC) que permite la comunicación entre una estación en tierra como QGroundControl y el vehículo [24].
- PX4 lleva configurado además un sistema de seguridad que permite resolver situaciones de peligro como batería baja, pérdida del RC, pérdida de la posición, etc [24].

En la imagen 4.5 se puede apreciar la placa Pixhawk con todos los periféricos y el cableado montado y detallado:

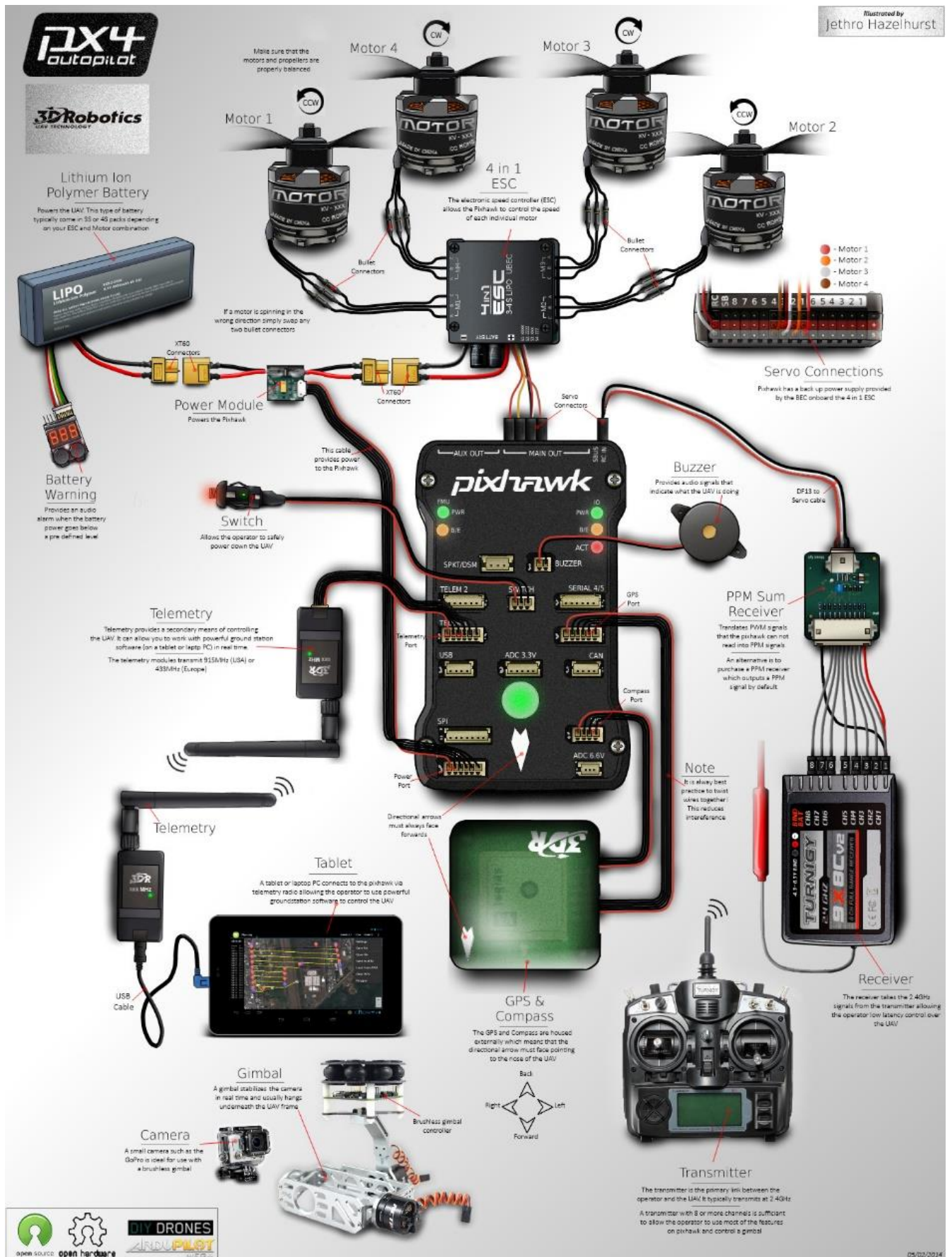


Fig. 4.5 Placa Pixhawk conectada a los periféricos [27]

La dirección en la que se oriente la placa determina la orientación de los ejes relativos al dron, lo cual es determinante a la hora de configurar el controlador, (véase Figgura 4.6) [29]:

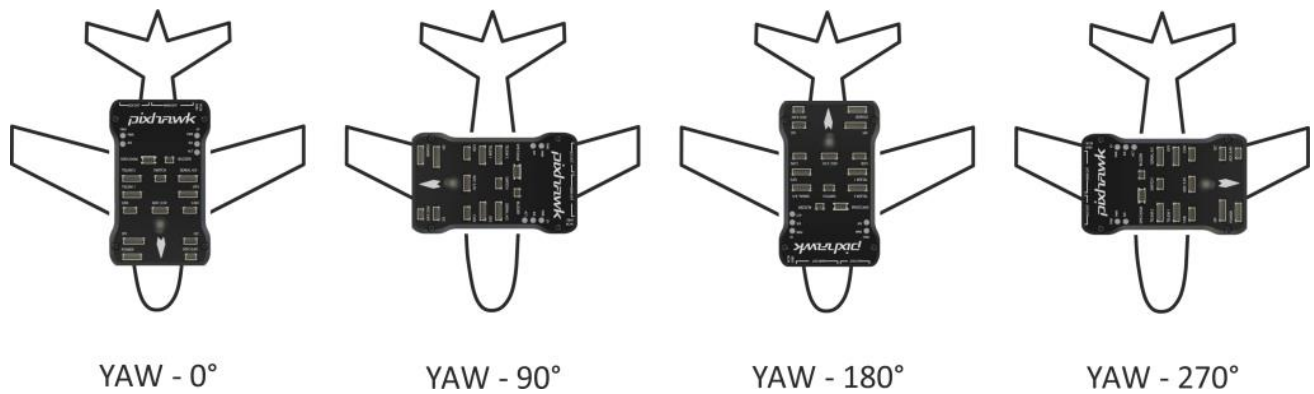


Fig. 4.6 Orientación Yaw en función de la orientación de la placa [29].

En el caso de que *yaw* esté orientado en  $0^\circ$  se obtienen los siguientes ejes relativos [24]:

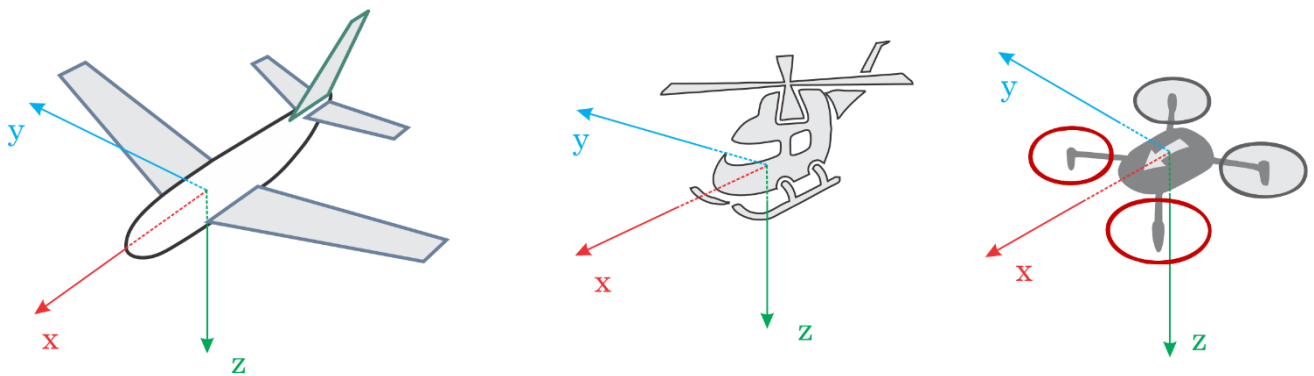


Fig. 4.7 Orientación de los ejes de coordenadas cuando Yaw es  $0^\circ$  [24].

## 4.2 Simulaciones

Para las simulaciones virtuales se ha creado un controlador que simula también el sistema de visualización. Para ello, se le han introducido unos parámetros iniciales y se evalúa la nueva posición cada 0,2 segundos. De esta manera se puede obtener la nueva posición del dron que determinará una nueva velocidad de salida y, en definitiva, un vuelo aproximado al que se tendría en la realidad con el sistema de visualización implementando.

El objetivo de estas simulaciones consiste en evaluar la respuesta de las distintas coordenadas por separado ante funciones de membresía que adopten forma triangular o trapezoidal, comparar estos resultados y seleccionarlos para comprobar cómo funcionan en conjunto y determinar si, realizando cambios poco sustanciales, son aptos para el aterrizaje o no.

Para ello se ha acotado el problema siguiendo las premisas que se dictan a continuación:

1. El dron partirá de las condiciones máximas de operación, esto es, 4 metros para los ejes X, Y y Z,  $45^\circ$  o  $-45^\circ$  para el ángulo de inclinación  $\alpha$  y  $45^\circ$  para el ángulo de inclinación  $\beta$ .
2. El dron partirá desde una posición y obtendrá los datos iniciales sin necesidad de buscarlos y siempre tendrá visualizada la posición de la plataforma objetivo.
3. Se tendrá como velocidad máxima para los ejes cartesianos 3 m/s y para la velocidad angular 0,5rad/s.
4. El dron no necesitará rotar respecto al eje Z puesto que según la segunda premisa siempre tendrá visualizada la plataforma.
5. El espacio solución para los parámetros en las distintas coordenadas se definirá con 4 o 5 reglas difusas.
6. Los ejes X y Z se comportarán de manera similar, es decir, se deberá evitar pasar el 0 (especialmente en el caso de la Z, en el que en posiciones muy próximas a la plataforma estar

por debajo del 0 podría indicar una colisión). El eje Y, sin embargo, podrá obtener valores tanto positivos como negativos.

7. Cuando en los tres ejes se alcanzase una posición que se considera adecuada y esté cerca del 0 se considerará que el dron ha llegado al objetivo. A continuación, se pararán los motores y caerá en la plataforma.

#### 4.2.1 Comparación entre funciones de membresía trapezoidal y triangular para los ejes X y Z:

Las figuras 4.8 y 4.9 presentan las funciones de membresía para Z en disposición trapezoidal y triangular, respectivamente:

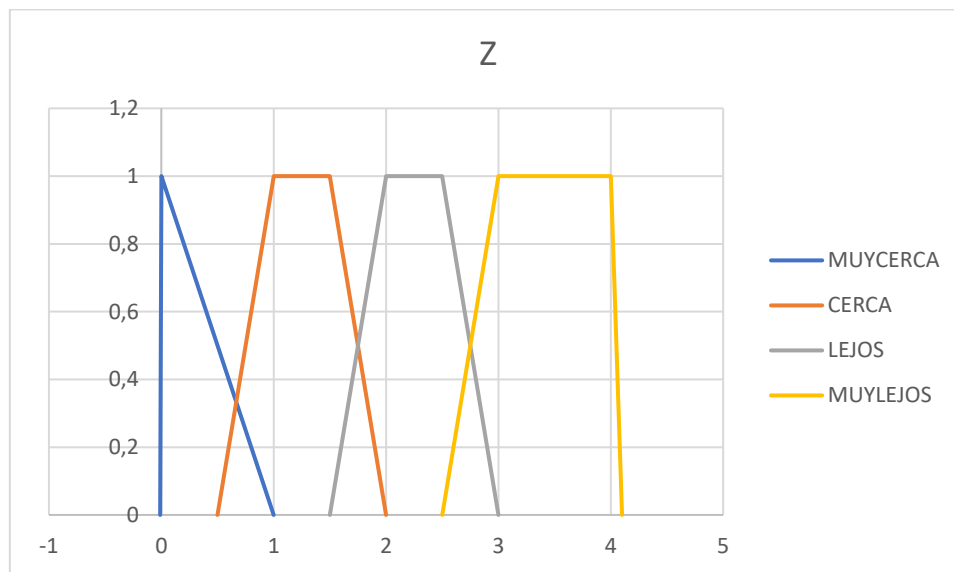


Fig. 4.8 Funciones de membresía para Z en disposición trapezoidal

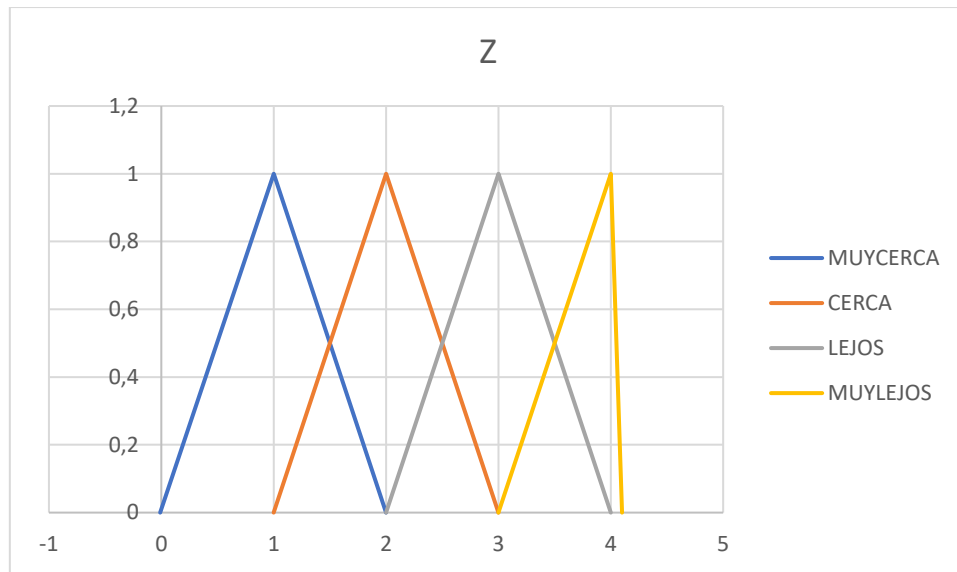


Fig. 4.9 Funciones de membresía para Z en disposición triangular

Las figuras 4.10 y 4.11 presentan funciones de membresía para VelZ en disposición triangular y trapezoidal, respectivamente:

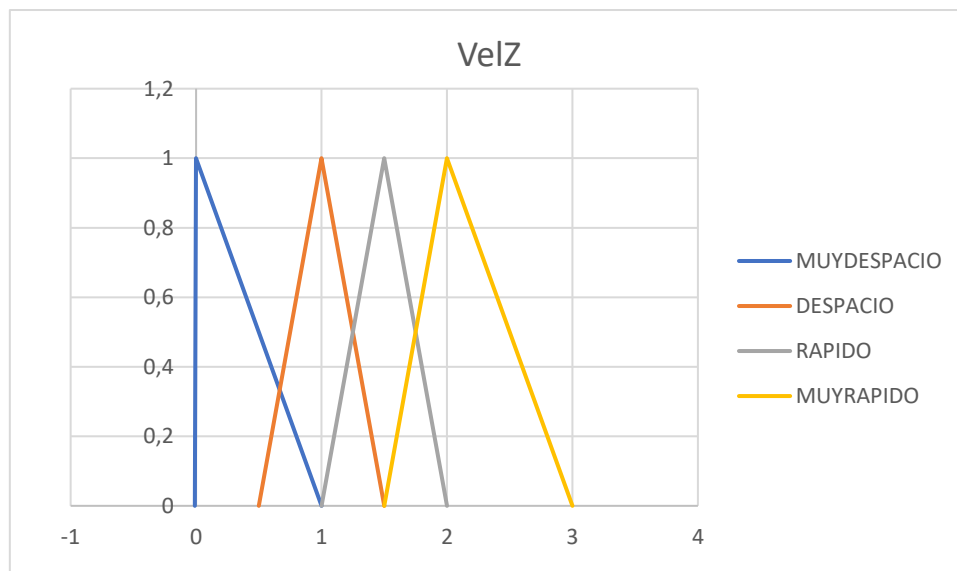


Fig. 4.10 Funciones de membresía para VelZ en disposición triangular

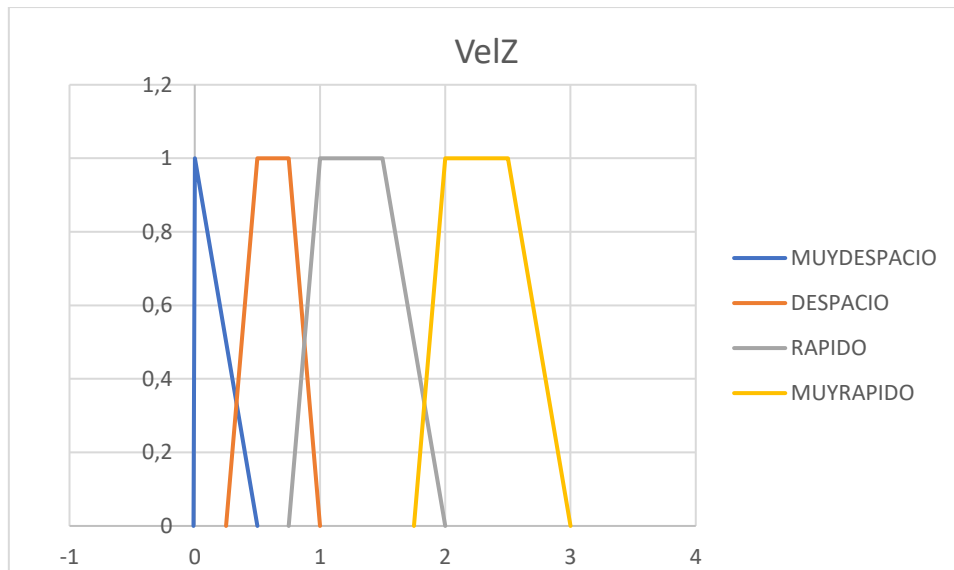


Fig. 4.11 Funciones de membresía para VelZ en disposición trapezoidal

La Tabla 1 presenta las reglas difusas para Z:

Tabla 1 Reglas difusas para Z

SI	ES	ENTONCES	ES
Z	MUYCERCA	VelZ	MUYDESPACIO
	CERCA		DESPACIO
	LEJOS		RAPIDO
	MUYLEJOS		MUYRAPIDO

En las Tablas 2-4 y las figuras 4.12-4.15 se presentan los resultados obtenidos, teniendo en cuenta que la velocidad será 0 en estos ejes cuando el dron se encuentre en un área de proximidad adecuada, que en este apartado se ha establecido como 2cm:

Tabla 2 Datos Z y VelZ combinación trapecio-triángulo

t	Trapecio	Triángulo
	Z	VelZ
0	4	2,25
0,2	3,55	2,25
0,4	3,1	2,25
0,6	2,65	1,8425
0,8	2,28149	1,5
1	1,98149	1,46609
1,2	1,68827	1,208
1,4	1,44668	1
1,6	1,24668	1
1,8	1,04668	1
2	0,846676	0,878839
2,2	0,670908	0,750068
2,4	0,520894	0,545532
2,6	0,411788	0,495
2,8	0,312788	0,495
3	0,213788	0,495
3,2	0,114788	0,495
3,4	0,0157878	0
tiempo total		3,4

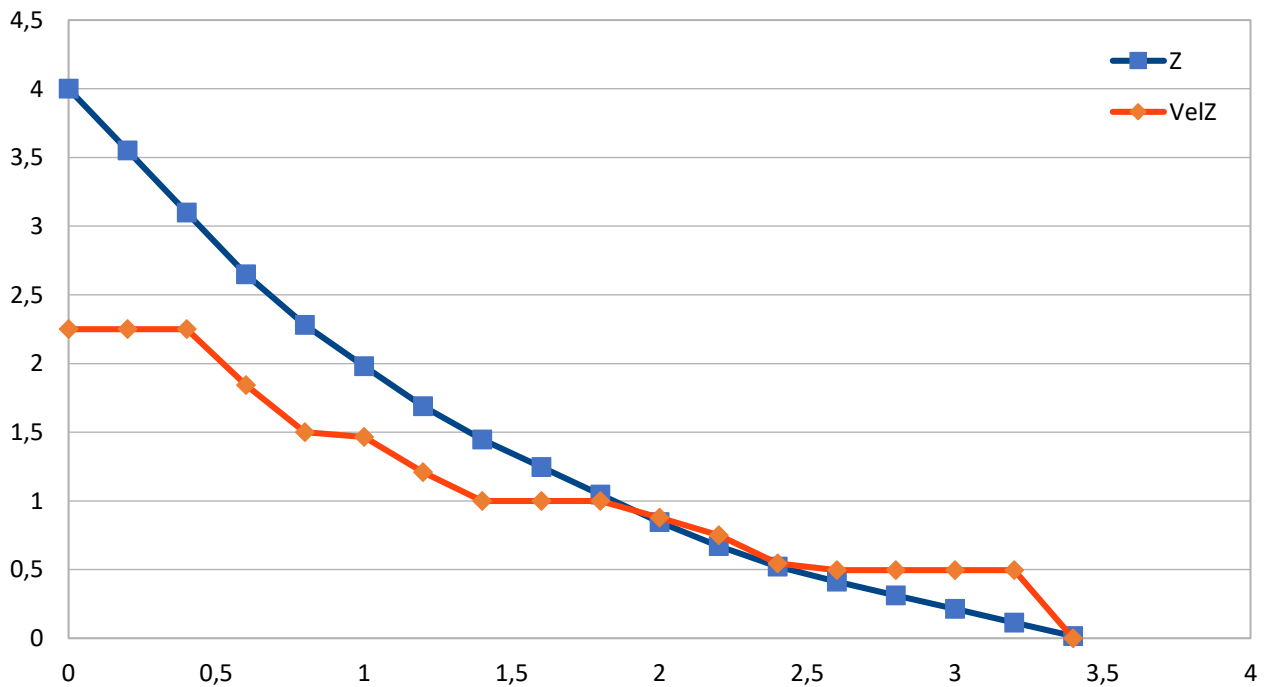


Fig. 4.12 Gráfico comparativo para Z y VelZ con combinación trapecio-triángulo



Tabla 3 Datos Z y VelZ combinación triángulo-triángulo

	Triángulo	Triángulo		Triángulo	Triángulo
t	Z	VelZ	t	Z	VelZ
0	4	2,25	2,4	1,08671	0,566708
0,2	3,55	1,97376	2,6	0,973364	0,495
0,4	3,15525	1,72924	2,8	0,874364	0,495
0,6	2,8094	1,36822	3	0,775364	0,495
0,8	2,5357	1,26194	3,2	0,676364	0,495
1	2,28337	1,17297	3,4	0,577364	0,495
1,2	2,04877	1,04354	3,6	0,478364	0,495
1,4	1,84006	0,881962	3,8	0,379364	0,495
1,6	1,66367	0,803451	4	0,280364	0,495
1,8	1,50298	0,747248	4,2	0,181364	0,495
2	1,35353	0,695533	4,4	0,082364	0,495
2,2	1,2443	0,638602	4,6	-0,016636	0
				tiempo total	4,6

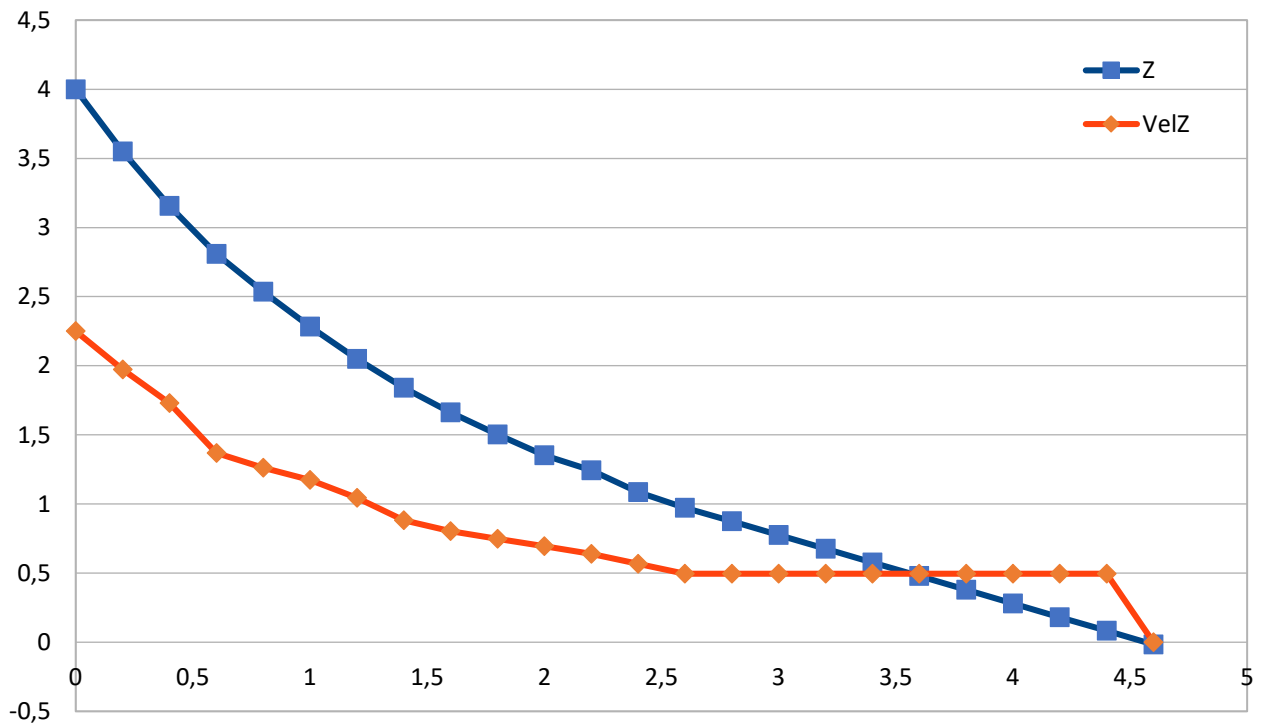


Fig.4.13 Gráfico comparativo para Z y VelZ con combinación triángulo-triángulo

Tabla 4 Datos Z y VelZ combinación trapecio-trapecio

Trapecio			Trapecio		
t	Z	VelZ	t	Z	VelZ
0	4	2,375	2,8	0,654544	0,468404
0,2	3,525	2,375	3	0,560863	0,371857
0,4	3,05	2,375	3,2	0,486491	0,245
0,6	2,575	1,55948	3,4	0,437491	0,245
0,8	2,2631	1,375	3,6	0,388491	0,245
1	1,9881	1,35992	3,8	0,339491	0,245
1,2	1,71612	1,05748	4	0,290491	0,245
1,4	1,50462	0,641974	4,2	0,241491	0,245
1,6	1,37623	0,625	4,4	0,192491	0,245
1,8	1,25123	0,625	4,6	0,143491	0,245
2	1,12623	0,625	4,8	0,0944915	0,245
2,2	1,00123	0,625	5	0,0454915	0,245
2,4	0,876228	0,578315	5,2	-0,0035085	0
2,6	0,760565	0,530105		tiempo total	5,2

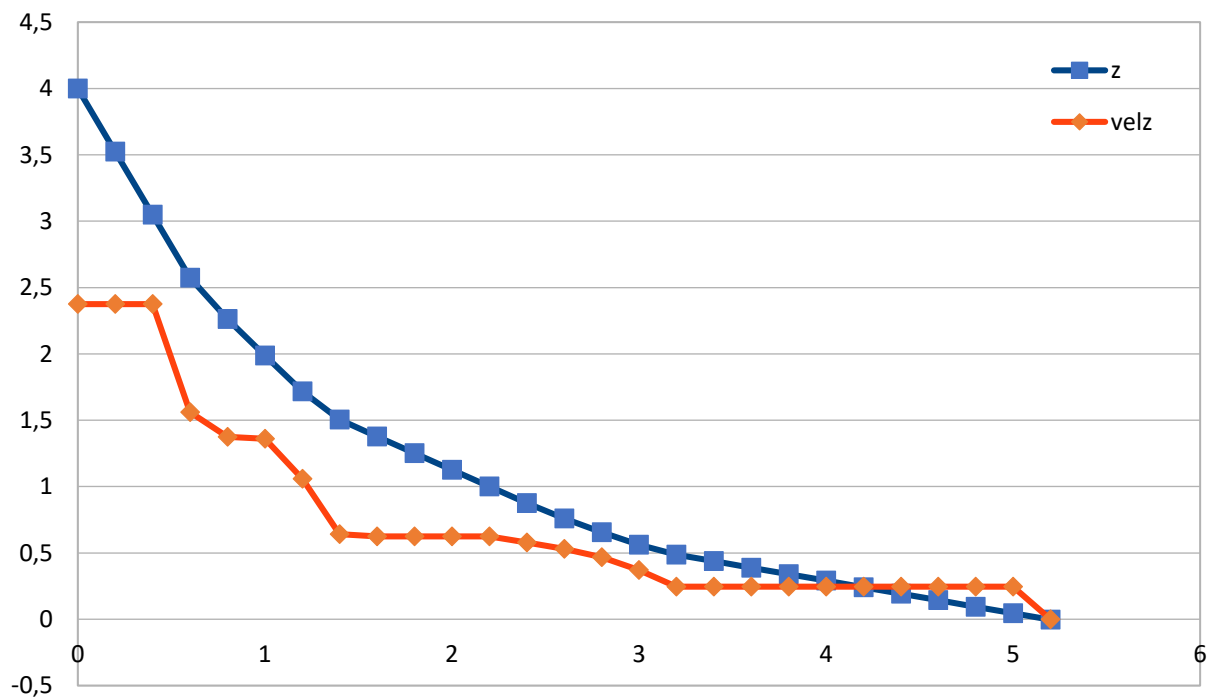


Fig.4.14 Gráfico comparativo para Z y VelZ con combinación trapecio-trapecio

Tabla 5 Datos Z y VelZ combinación triángulo-trapecio

Triángulo			Trapezio			Triángulo			Trapezio		
t	Z	VelZ	t	Z	VelZ	t	Z	VelZ	t	Z	VelZ
0	4	2,375	2,6	1,34152	0,435723	5,2	0,58554	0,245			
0,2	3,525	1,8956	2,8	1,25437	0,405611	5,4	0,53654	0,245			
0,4	3,14588	1,55505	3	1,17325	0,370817	5,6	0,48754	0,245			
0,6	2,83487	1,27739	3,2	1,09909	0,329187	5,8	0,43854	0,245			
0,8	2,57939	1,14107	3,4	1,03325	0,278541	6	0,38954	0,245			
1	2,35118	1,0055	3,6	0,97754	0,245	6,2	0,34054	0,245			
1,2	2,15008	0,836263	3,8	0,92854	0,245	6,4	0,29154	0,245			
1,4	1,98283	0,618457	4	0,87954	0,245	6,6	0,24254	0,245			
1,6	1,85913	0,57681	4,2	0,83054	0,245	6,8	0,19354	0,245			
1,8	1,74377	0,543841	4,4	0,78154	0,245	7	0,14454	0,245			
2	1,635	0,515401	4,6	0,73254	0,245	7,2	0,0955398	0,245			
2,2	1,53192	0,489041	4,8	0,68354	0,245	7,4	0,0465398	0,245			
2,4	1,43412	0,462998	5	0,63454	0,245	7,6	-0,0024602	0			
							tiempo total				7,6

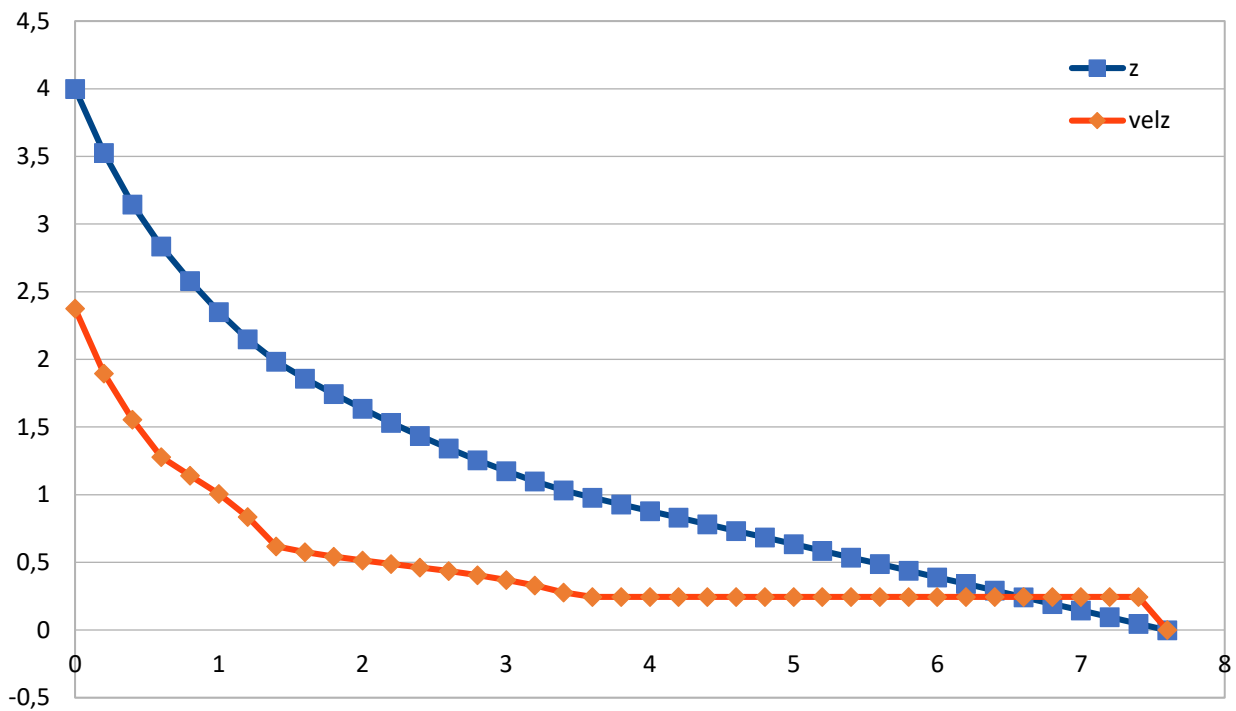


Fig.4.15 Gráfico comparativo para Z y VelZ para combinación triángulo-trapecio

Se puede apreciar cómo las funciones de membresía que mejor responden a los parámetros X y Z son para la distancia en trapezio y para la velocidad en triángulo. Aparte es destacable que las otras combinaciones, pese a que presentan cambios en la velocidad más suaves, terminan en valores negativos lo que indica una colisión contra la plataforma objetivo. Este suceso es del todo indeseable, ya que dependiendo de la velocidad de aproximación puede ocasionar daños a la aeronave.

#### 4.2.2 Comparación entre funciones de membresía trapezoidal y triangular para el eje Y:

Las figuras 4.16 y 4.17 presentan las funciones de membresía para Y en disposición trapezoidal y triangular, respectivamente:

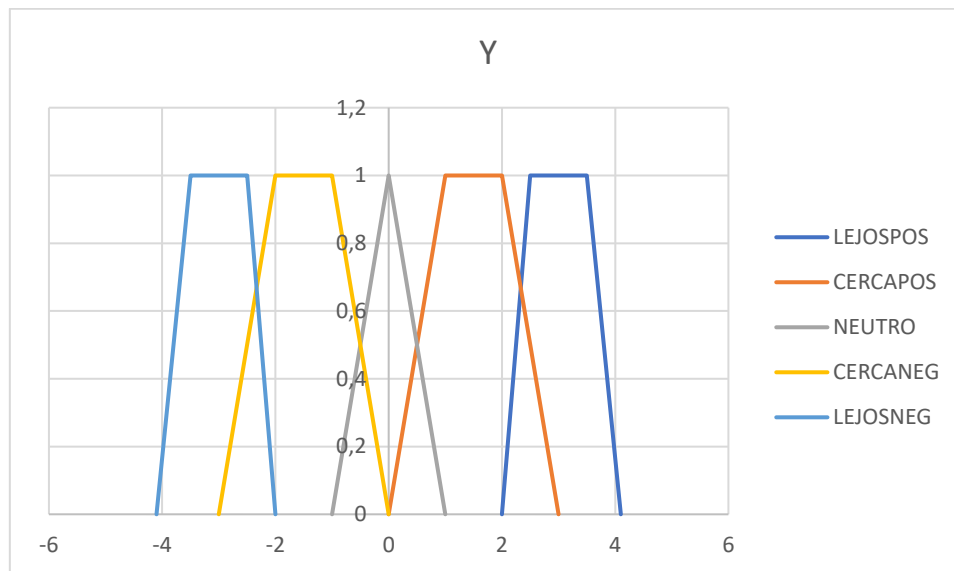


Fig. 4.16 Funciones de membresía para Y con disposición trapezoidal

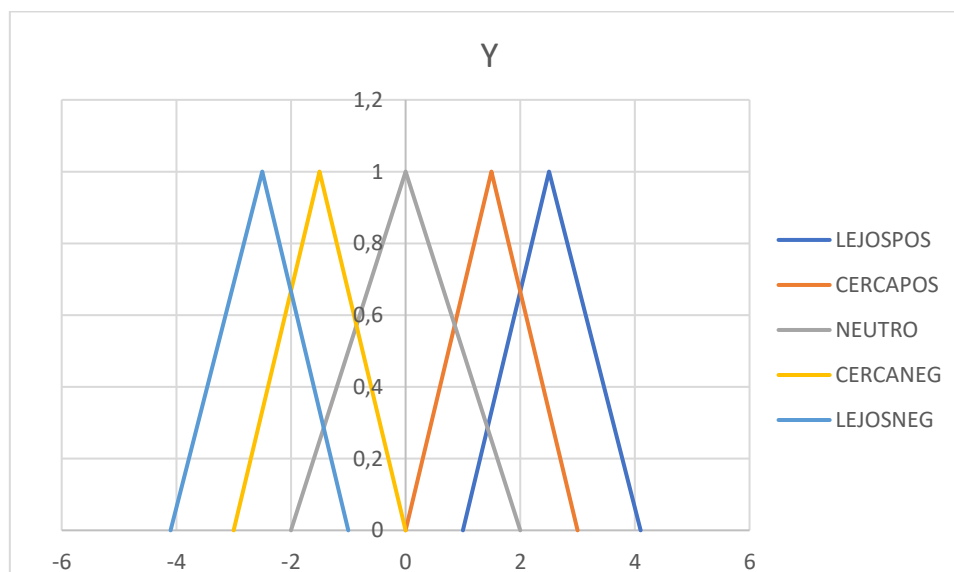


Fig. 4.17 Funciones de membresía para Y con disposición triangular

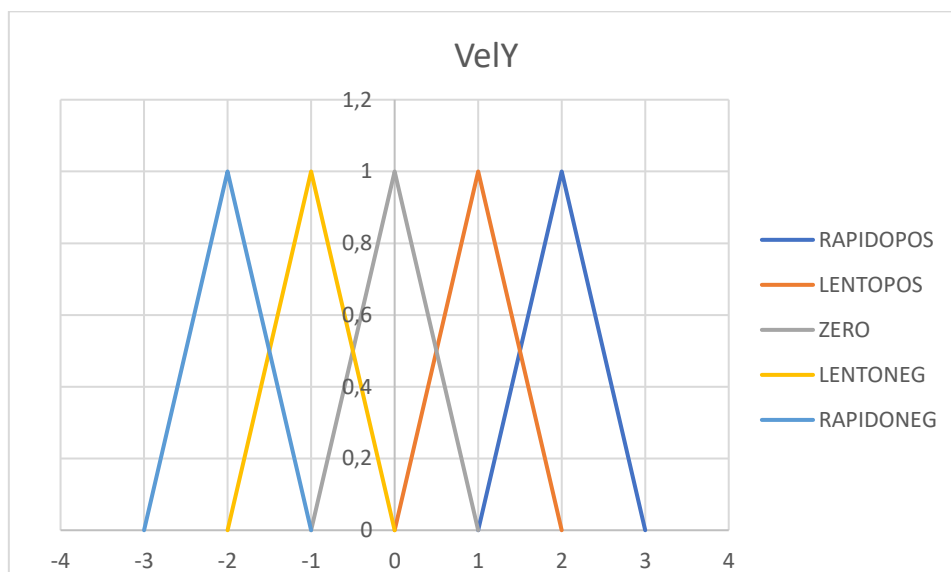


Fig. 4.18 Funciones de membresía para  $VelY$  con disposición triangular

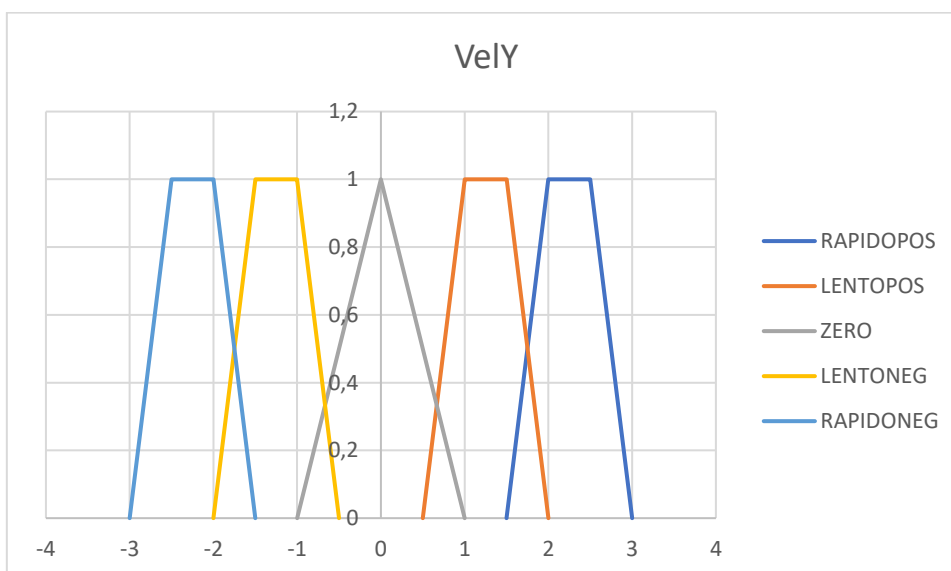


Fig. 4.19 Funciones de membresía para  $VelY$  con disposición trapezoidal

La Tabla 6 presenta las reglas difusas para Y:

Tabla 6 Reglas difusas para Y

SI	ES	ENTONCES	ES
Y	LEJOSPOS	VelY	RAPIDOPOS
	CERCAPOS		LENTOPOS
	NEUTRO		ZERO
	CERCANEG		LENTONEG
	LEJOSNEG		RAPIDONEG

En las Tablas 7-10 y las figuras 4.20-4.23 se presentan los resultados obtenidos. Al igual que en el apartado anterior se ha decidido que la velocidad será 0 cuando el dron se encuentre a menos de 2cm de la plataforma del eje Y:

Tabla 7 Datos Y y VelY combinación trapecio-trapecio

	Trapecio	Trapecio
t	Y	VelY
0	4	2,25
0,2	3,55	2,25
0,4	3,1	2,25
0,6	2,65	1,93318
0,8	2,26336	1,68863
1	1,92564	1,25
1,2	1,67564	1,25
1,4	1,42564	1,25
1,6	1,17564	1,25
1,8	0,925638	1,08776
2	0,708086	0,774242
2,2	0,553238	0,6172
2,4	0,429798	0,504836
2,6	0,328831	0,412438
2,8	0,246343	0,331615
3	0,18002	0,259766
3,2	0,128067	0,196895
3,4	0,088688	0,143925
3,6	0,0599029	0,101501
3,8	0,0396027	0,0693199
4	0,0257387	0,0461115
4,2	0,0165164	0
	tiempo total	4,2

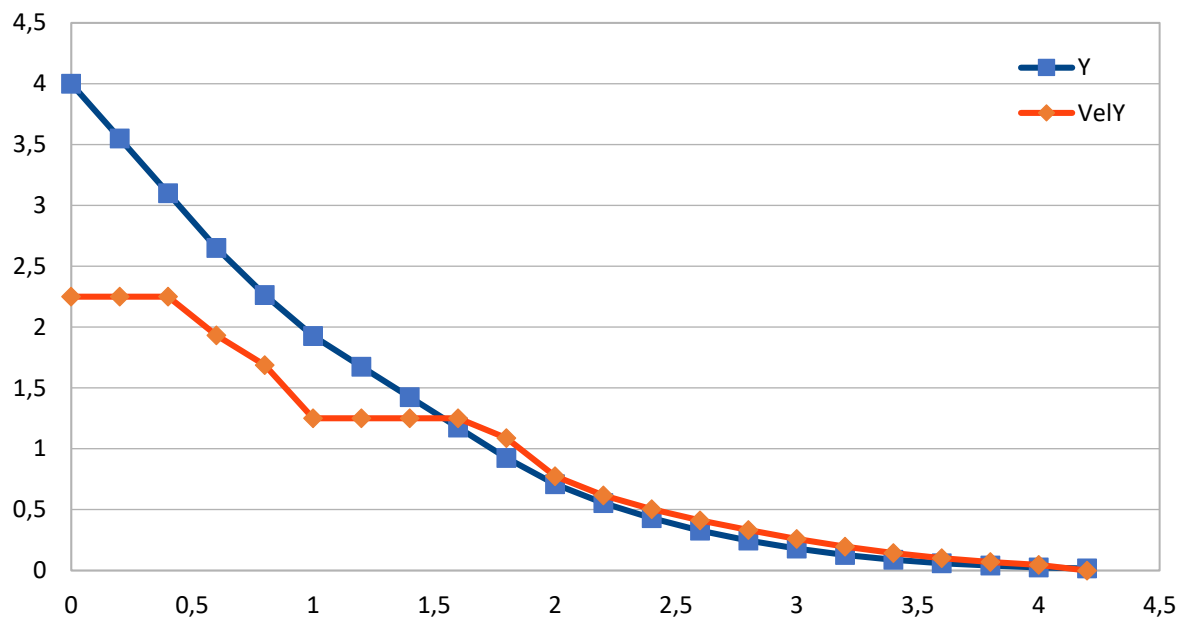


Fig. 4.20 Gráfico comparativo para Y y VelY con combinación trapecio-trapecio

Tabla 8 Datos Y y VelY combinación trapecio-triángulo

t	Trapecio	Triángulo
	Y	VelY
0	4	2
0,2	3,6	2
0,4	3,2	2
0,6	2,8	1,73529
0,8	2,45294	1,55495
1	2,14195	1,3321
1,2	1,87553	1
1,4	1,67553	1
1,6	1,47553	1
1,8	1,27553	1
2	1,07553	1
2,2	0,875531	0,808329
2,4	0,713865	0,651836
2,6	0,583498	0,556187
2,8	0,47226	0,481488
3	0,375963	0,415577
3,2	0,292847	0,353517
3,4	0,222144	0,293506
3,6	0,163443	0,235714
3,8	0,1163	0,181721
4	0,0799555	0,133829
4,2	0,0531898	0,0940751
4,4	0,0343748	0,0633616
4,6	0,0217025	0,0411851
4,8	0,0134654	0
tiempo total		4,8

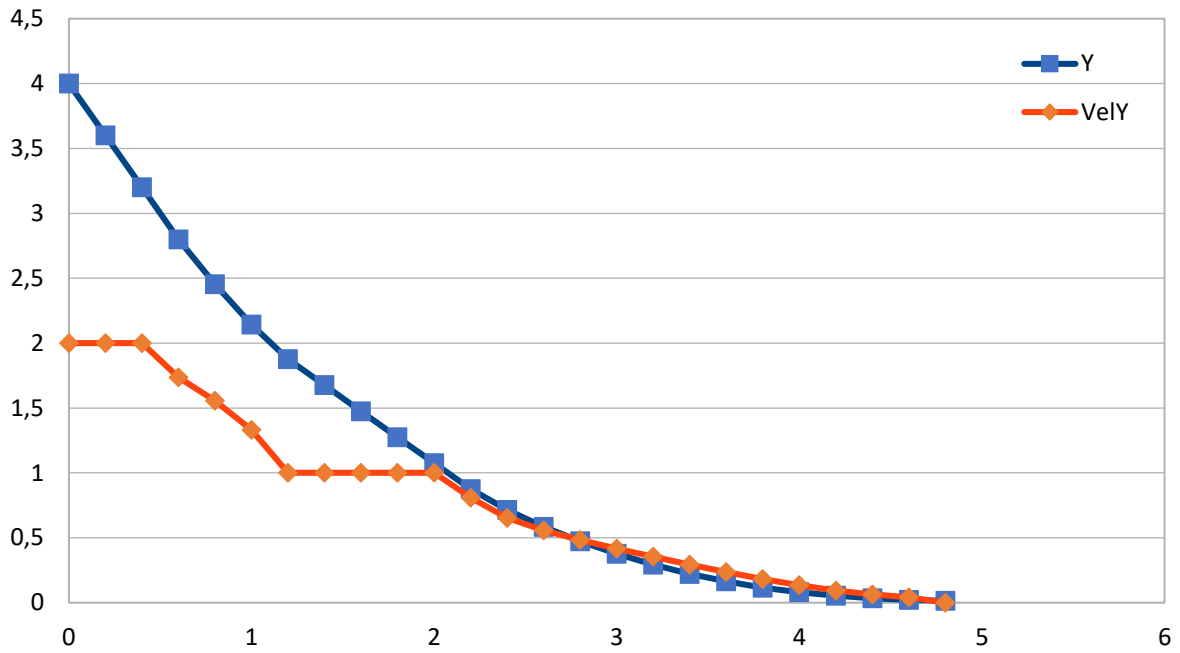


Fig. 4.21 Gráfico comparativo para Y y VelY con combinación trapecio-triángulo

Tabla 9 Datos Y y VelY combinación triángulo-triángulo

	Triángulo	Triángulo		Triángulo	Triángulo
t	Y	VelY	t	Y	VelY
0	4	2	3	0,521099	0,381159
0,2	3,6	2	3,2	0,444867	0,347043
0,4	3,2	2	3,4	0,375459	0,312222
0,6	2,8	1,79494	3,6	0,313014	0,277035
0,8	2,44101	1,62213	3,8	0,257607	0,24201
1	2,11659	1,52935	4	0,209205	0,207832
1,2	1,81072	1,3159	4,2	0,167639	0,175272
1,4	1,54754	1,0979	4,4	0,132584	0,145089
1,6	1,32796	0,91234	4,6	0,103567	0,117917
1,8	1,14549	0,729331	4,8	0,0799832	0,094176
2	0,999621	0,542263	5	0,061148	0,0740257
2,2	0,891168	0,510326	5,2	0,0463428	0,0573762
2,4	0,789103	0,478719	5,4	0,0348676	0,0439439
2,6	0,69336	0,446877	5,6	0,0260788	0,0333264
2,8	0,603984	0,414425	5,8	0,0194135	0
				tiempo total	5,8



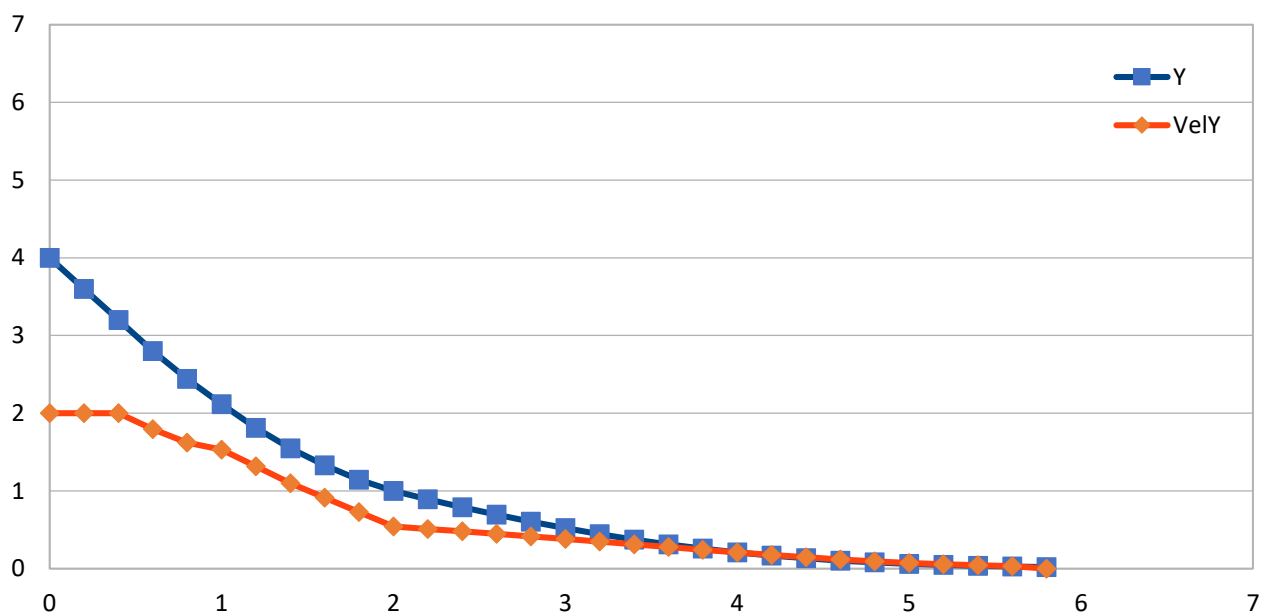


Fig. 4.22 Gráfico comparativo para Y y VelY con combinación triángulo-triángulo

Tabla 10 Datos Y y VelY combinación triángulo-trapecio

t	Triángulo	Trapecio	t	Triángulo	Trapecio
	Y	VelY		Y	VelY
0	4	2,25	2,8	0,439729	0,367801
0,2	3,55	2,25	3	0,366169	0,322727
0,4	3,1	2,25	3,2	0,301623	0,279469
0,6	2,65	1,99615	3,4	0,245729	0,238568
0,8	2,25077	1,8412	3,6	0,198016	0,200617
1	1,88253	1,59082	3,8	0,157892	0,166161
1,2	1,56437	1,25449	4	0,12466	0,1356
1,4	1,31347	1,01584	4,2	0,0975401	0,109128
1,6	1,1103	0,787118	4,4	0,0757144	0,0867195
1,8	0,952877	0,616019	4,6	0,0583705	0,0681485
2	0,829674	0,562191	4,8	0,0447408	0,0530468
2,2	0,717235	0,511186	5	0,0341315	0,0409651
2,4	0,614998	0,46205	5,2	0,0259385	0,0314309
2,6	0,522588	0,414297	5,4	0,0196523	0
				tiempo total	5,4

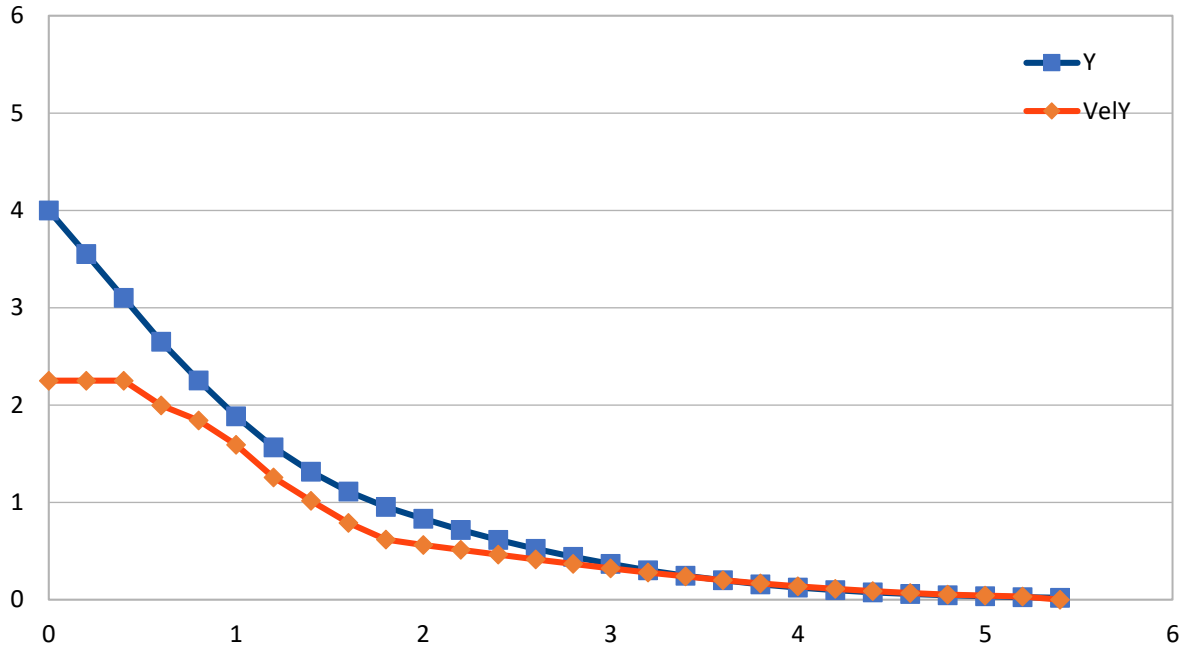


Fig. 4.23 Gráfico comparativo para Y y VelY con combinación triángulo-trapecio

De los resultados presentados, se deduce que en esta prueba la mejor combinación posible para los parámetros en Y resultan de la combinación de dos funciones de membresía con forma trapezoidal con un tiempo de 4,2 segundos. Cabe destacar que el tiempo de vuelo en la coordenada Y es mayor que en las coordenadas X y Z ( $t=3,4$ ), esto nos indica que pese a haber impuesto una regla más que en el test anterior no es suficiente para igualar la rapidez de aproximación. Sin embargo, en las gráficas se puede apreciar cómo la velocidad tiene una variación progresiva y gradual, similar a la curva del desplazamiento, además no presenta valores negativos en ninguna de las pruebas. Estas características implican que pese a los peores resultados temporales el sistema gana en calidad de aproximación.

#### 4.2.3 Comparación entre funciones de membresía trapezoidal y triangular para el ángulo $\alpha$ :

Las figuras 4.24 y 4.25 presentan las funciones de membresía para  $\alpha$  en disposición trapezoidal y triangular, respectivamente:

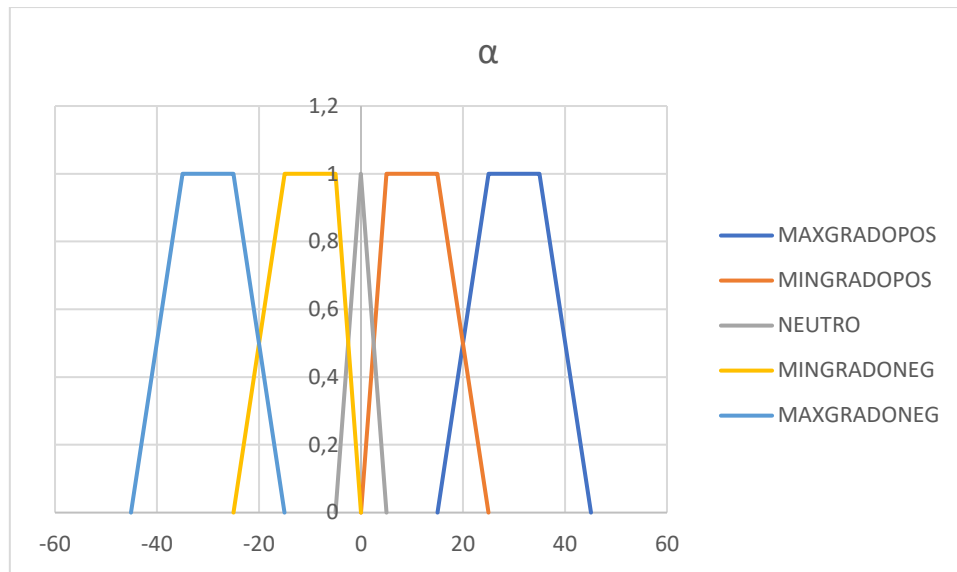


Fig. 4.24 Funciones de membresía para alfa con disposición trapezoidal

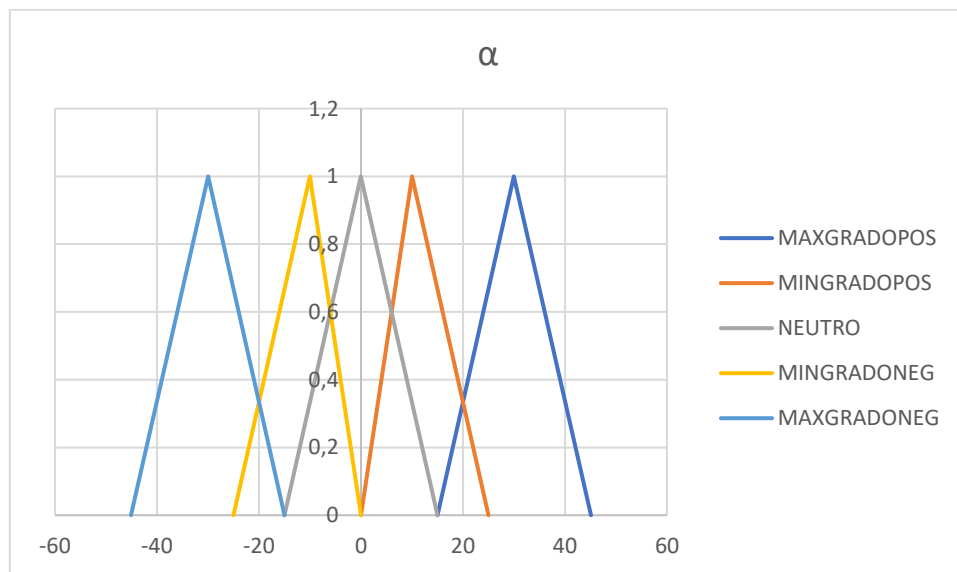


Fig. 4.25 Funciones de membresía para alfa con disposición triangular

Las figuras 4.26 y 4.27 presentan las funciones de membresía para  $W_x$  en disposición trapezoidal y triangular, respectivamente:

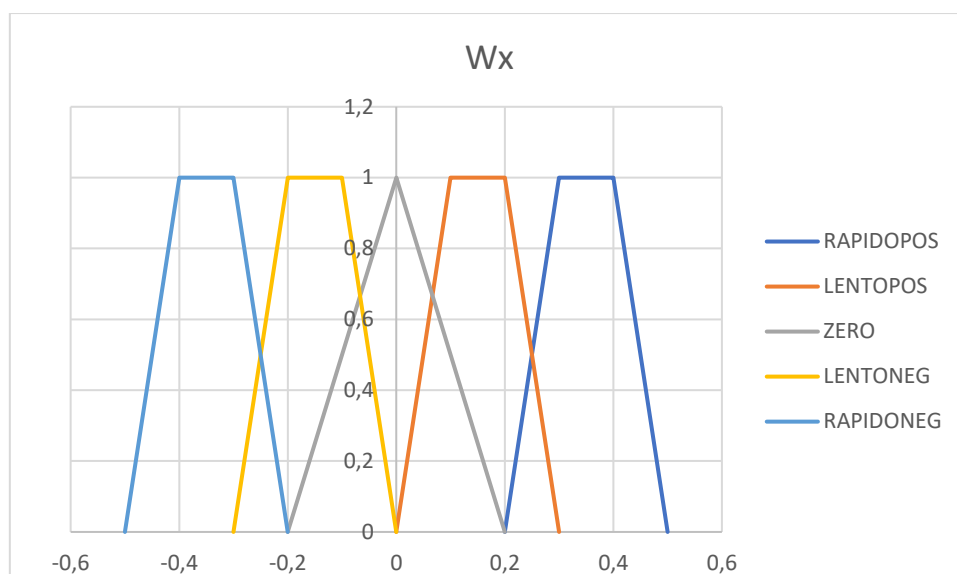


Fig. 4.26 Funciones de membresía para  $W_x$  con disposición trapezoidal

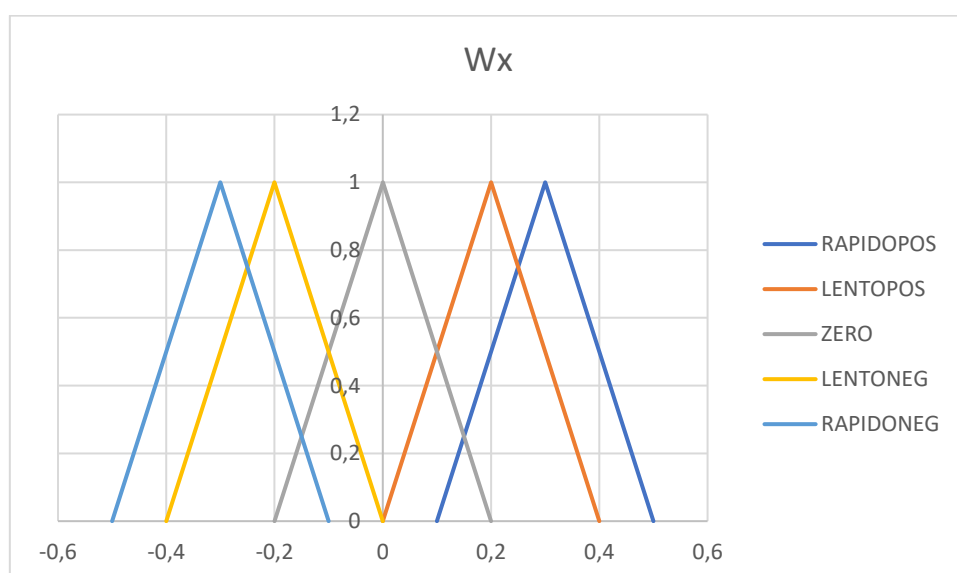


Fig. 4.27 Funciones de membresía para  $W_x$  con disposición triangular

La Tabla 11 presenta las reglas difusas para  $\alpha$ :

Tabla 11 Reglas difusas alfa

SI	ES	ENTONCES	ES
$\alpha$	MAXGRADOPOS	$W_x$	RAPIDOPOS
	MINGRADOPOS		LENTOPOS
	NEUTRO		ZERO
	MINGRADONEG		LENTONEG
	MAXGRADONEG		RAPIDONEG

En las Tablas 12-15 y las figuras 4.28-4.35 se presentan los resultados obtenidos. Se entiende como una inclinación adecuada  $\pm 0,5^\circ$  respecto de la plataforma objetivo:

Tabla 12 Datos alfa y Wx combinación trapecio-trapecio

	Trapecio	Trapecio
t	$\alpha$	Wx
0	-45	-0,35
0,2	-40,9893	-0,35
0,4	-36,9786	-0,35
0,6	-32,9679	-0,35
0,8	-28,9572	-0,35
1	-24,9465	-0,348407
1,2	-20,9541	-0,265377
1,4	-17,9131	-0,215404
1,6	-15,4447	-0,162609
1,8	-13,5814	-0,15
2	-11,8625	-0,15
2,2	-10,1437	-0,15
2,4	-8,42478	-0,15
2,6	-6,70591	-0,15
2,8	-4,98704	-0,149227
3	-3,27704	-0,0860861
3,2	-2,29056	-0,0636384
3,4	-1,56132	-0,0476097
3,6	-1,01576	-0,034288
3,8	-0,622848	-0,0230877
4	-0,358279	0
	tiempo total	4

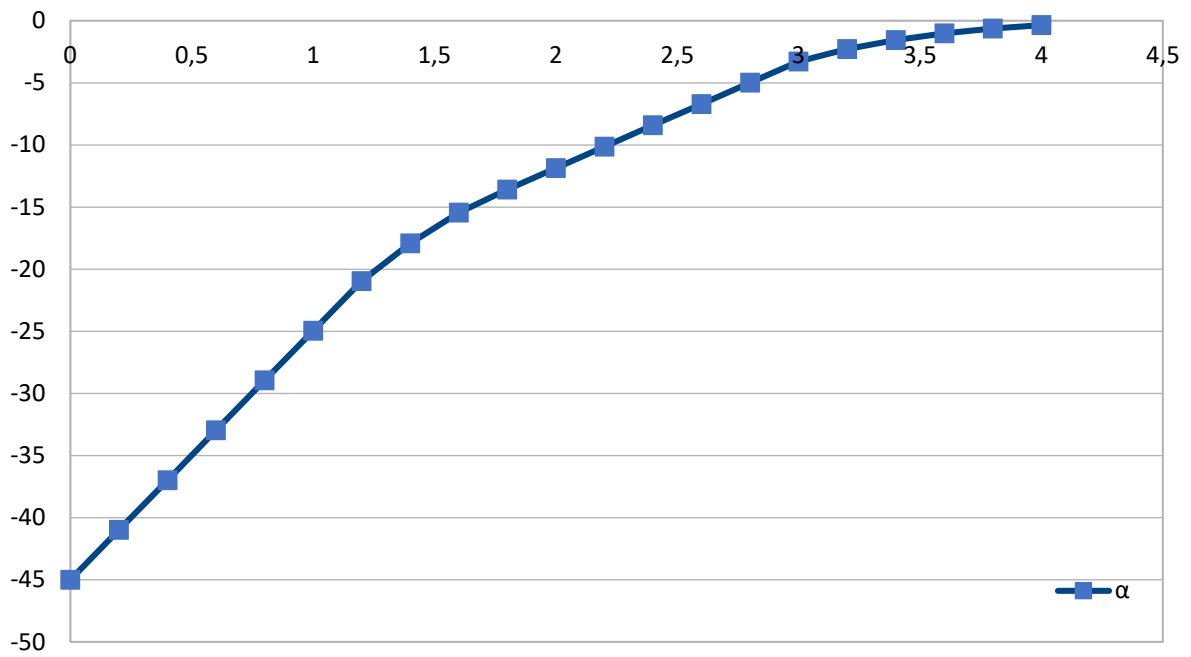


Fig. 4.28 Gráfico de la evolución de alfa con combinación trapecio-trapecio

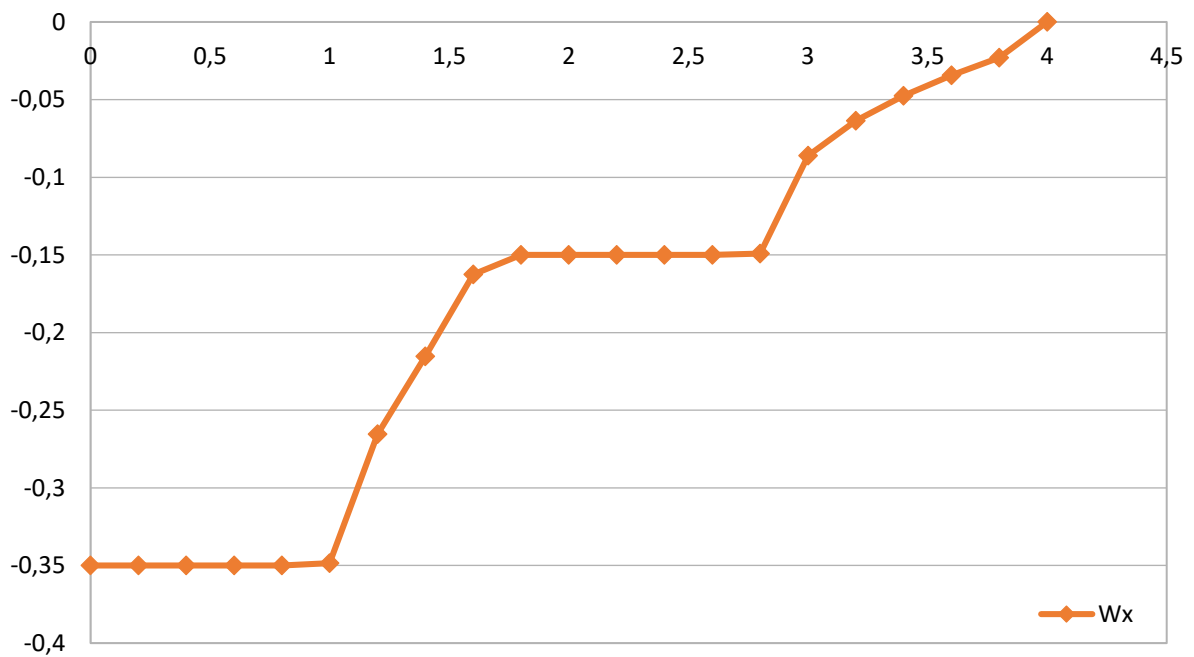


Fig. 4.29 Gráfico de la evolución de  $W_x$  con combinación trapecio-trapecio

Tabla 13 Datos alfa y Wx combinación trapecio-triángulo

	Trapecio	Triángulo
t	$\alpha$	Wx
0	-45	-0,3
0,2	-41,5623	-0,3
0,4	-38,1245	-0,3
0,6	-34,6868	-0,3
0,8	-31,249	-0,3
1	-27,8113	-0,3
1,2	-24,3736	-0,289139
1,4	-21,0603	-0,257176
1,6	-18,1133	-0,236795
1,8	-15,3998	-0,207278
2	-13,0246	-0,2
2,2	-10,7328	-0,2
2,4	-8,44093	-0,2
2,6	-6,1491	-0,2
2,8	-3,85728	-0,140138
3	-2,25142	-0,0933493
3,2	-1,18172	-0,0612547
3,4	-0,479846	0
	tiempo total	3,4

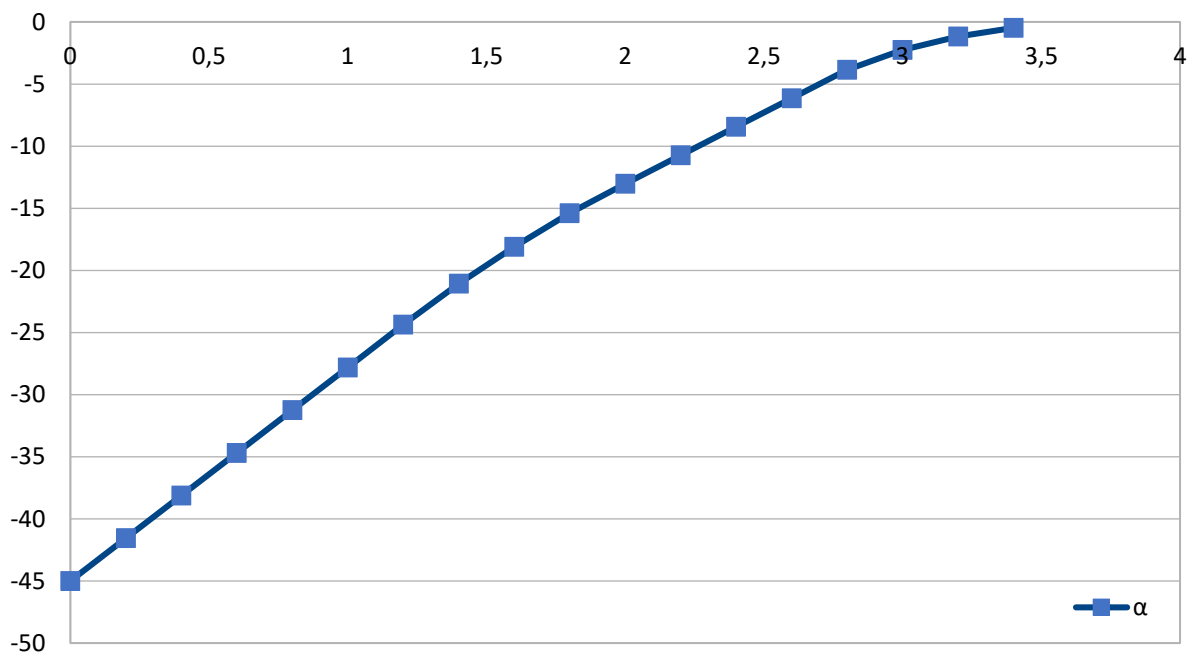


Fig. 4.30 Gráfico de la evolución de alfa con combinación trapecio-triángulo

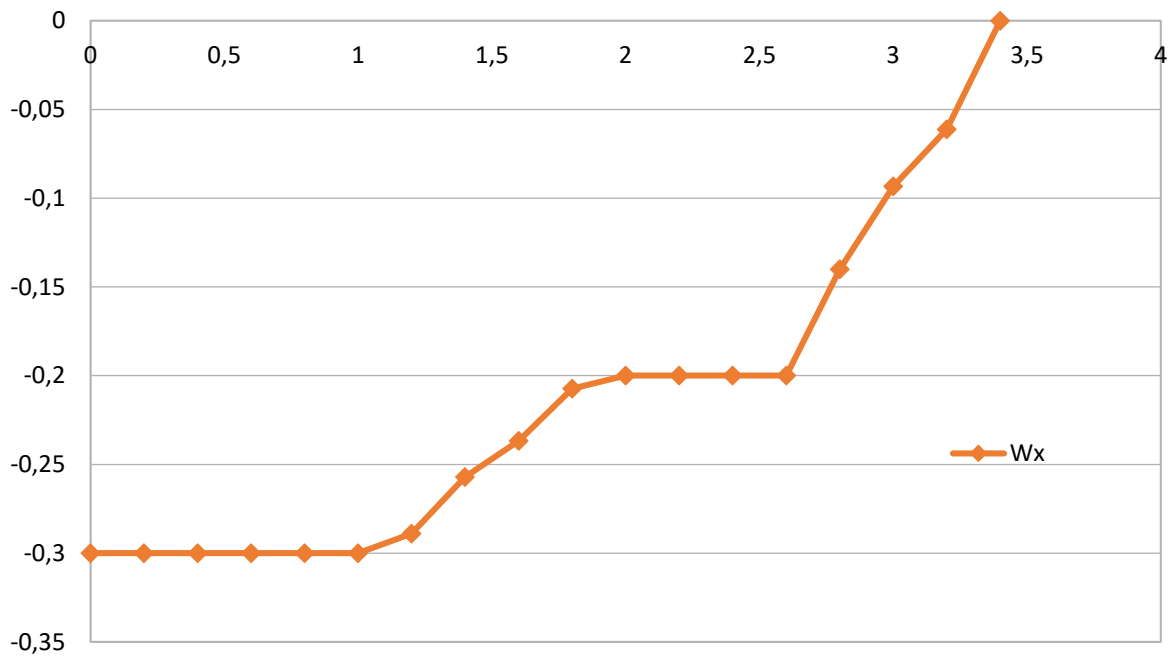


Fig. 4.31 Gráfico de la evolución de Wx con combinación trapecio-triángulo

Tabla 14 Datos alfa y Wx combinación triángulo-triángulo

t	Triángulo	Triángulo
	$\alpha$	Wx
0	-45	-0,3
0,2	-41,5623	-0,3
0,4	-38,1245	-0,3
0,6	-34,6868	-0,3
0,8	-31,249	-0,3
1	-27,8113	-0,3
1,2	-24,3736	-0,29131
1,4	-21,0354	-0,258355
1,6	-18,0749	-0,234129
1,8	-15,392	-0,205592
2	-13,0361	-0,159341
2,2	-11,2102	-0,138469
2,4	-9,62343	-0,125847
2,6	-8,18133	-0,115838
2,8	-6,85393	-0,106489
3	-5,63366	-0,0970274
3,2	-4,52181	-0,0869963
3,4	-3,52491	-0,0761361
3,6	-2,65245	-0,0644057
3,8	-1,91442	-0,0520674
4	-1,31777	-0,039757
4,2	-0,862192	-0,0284074
4,4	-0,536668	-0,0189369
4,6	-0,319665	0
tiempo total		4,6



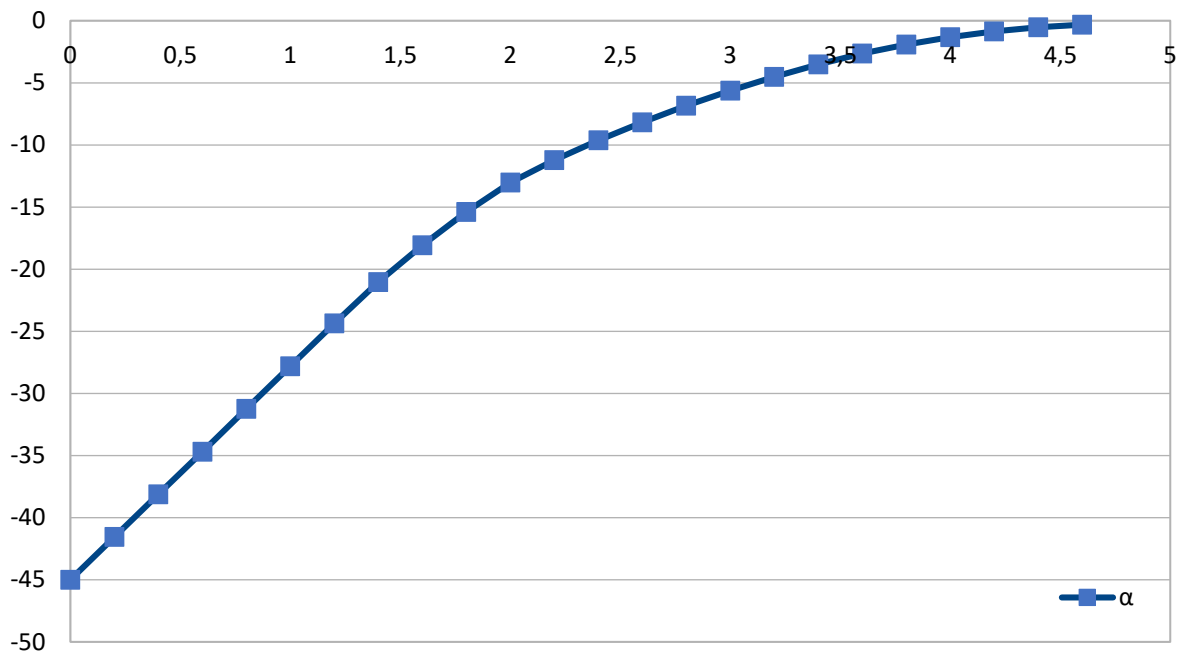


Fig. 4.32 Gráfico de la evolución de alfa con combinación triángulo-triángulo

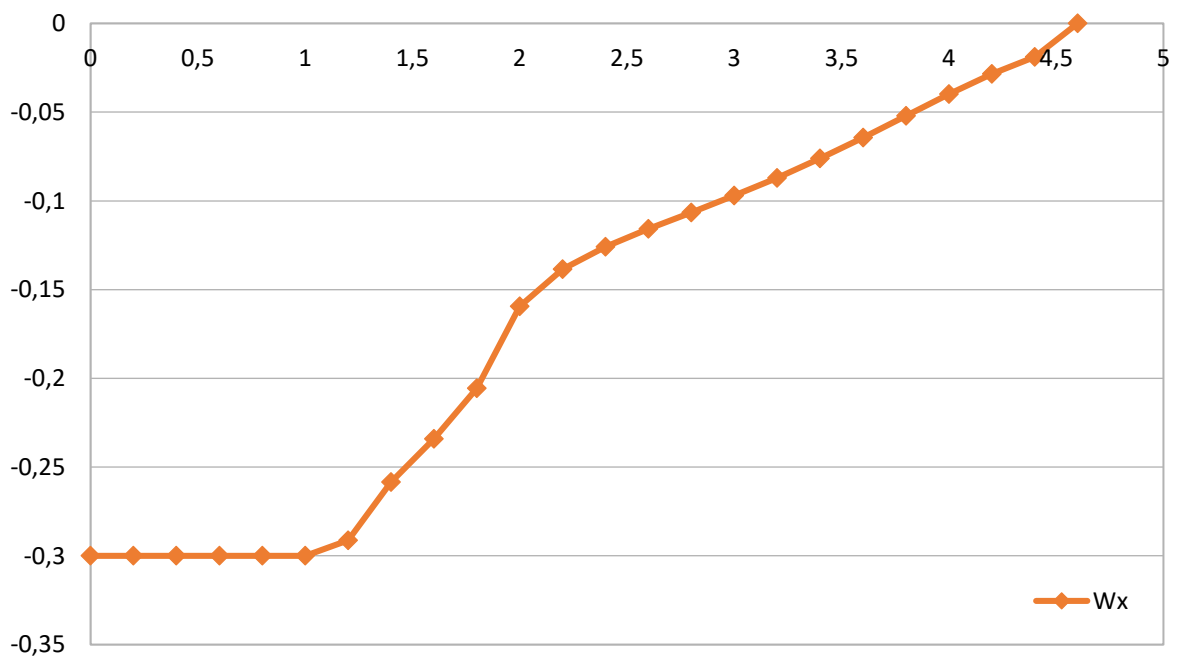


Fig. 4.33 Gráfico de la evolución de  $Wx$  con combinación triángulo-triángulo

Tabla 15 Datos alfa y Wx combinación triángulo-trapecio

	Triángulo	Trapezio		Triángulo	Trapezio
t	$\alpha$	Wx	t	$\alpha$	Wx
0	-45	-0,35	3	-6,76475	-0,0744924
0,2	-40,9893	-0,35	3,2	-5,91113	-0,0686045
0,4	-36,9786	-0,35	3,4	-5,12498	-0,062875
0,6	-32,9679	-0,35	3,6	-4,40449	-0,0572277
0,8	-28,9572	-0,35	3,8	-3,74871	-0,0516278
1	-24,9465	-0,348631	4	-3,1571	-0,0460765
1,2	-20,9515	-0,266727	4,2	-2,6291	-0,0406074
1,4	-17,8951	-0,212329	4,4	-2,16378	-0,0352822
1,6	-15,462	-0,161468	4,6	-1,75947	-0,0301829
1,8	-13,6117	-0,124219	4,8	-1,4136	-0,0254008
2	-12,1882	-0,109429	5	-1,12253	-0,0210224
2,2	-10,9343	-0,101033	5,2	-0,881635	-0,017116
2,4	-9,77652	-0,094826	5,4	-0,685501	-0,0137213
2,6	-8,6899	-0,0873341	5,6	-0,528266	-0,0108459
2,8	-7,68912	-0,0806672	5,8	-0,4039799	0
				tiempo total	5,8

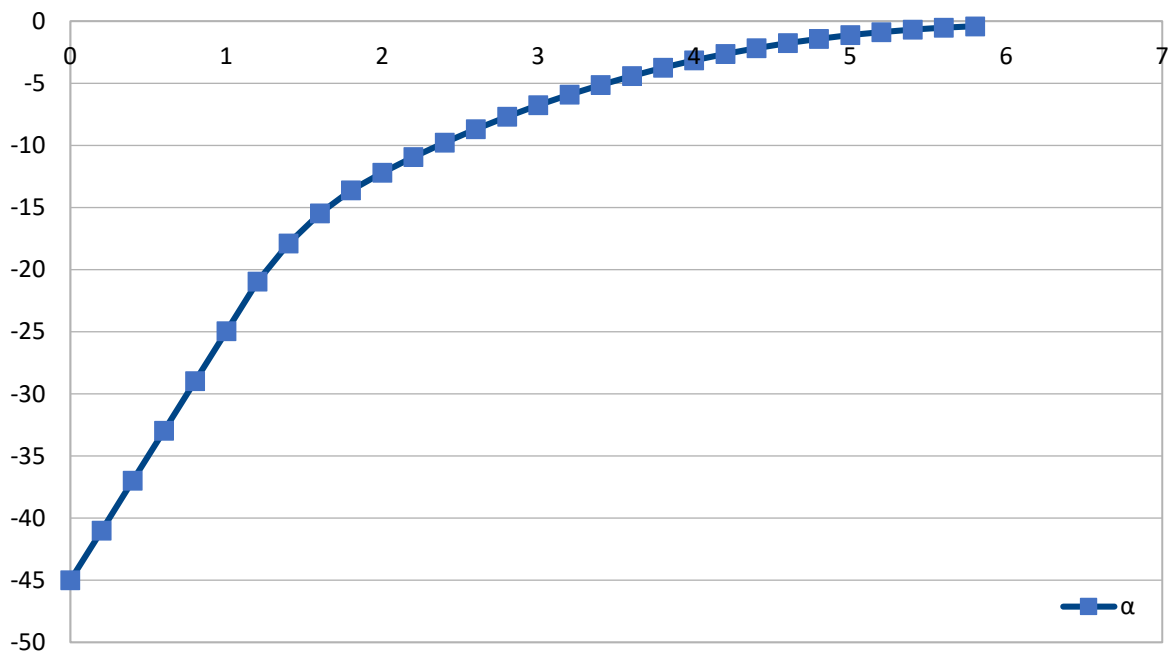


Fig. 4.34 Gráfico de la evolución de alfa con combinación triángulo-trapecio

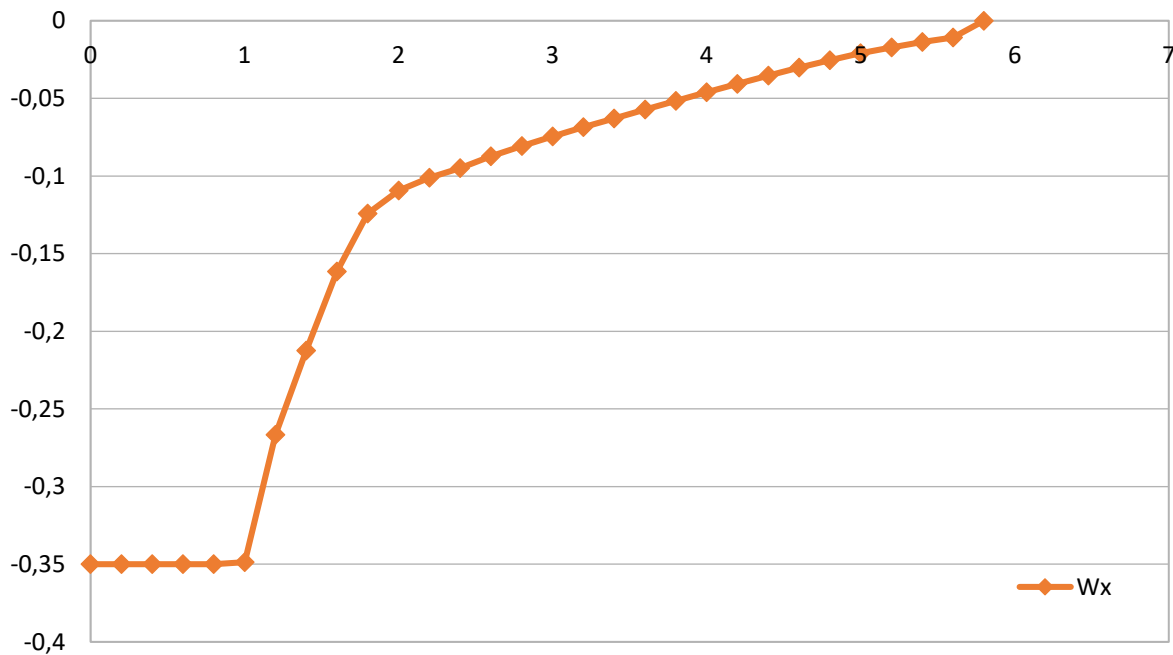


Fig. 4.35 Gráfico de la evolución de alfa con combinación triángulo-trapecio

En este experimento se ha tenido que separar la velocidad y la rotación por un problema de magnitudes. las magnitudes en grados que llegan hasta el valor -45 eran demasiado grandes en comparación a la velocidad angular.

De los resultados obtenidos se deduce que existen diferencias significativas respecto al tiempo de llegada al ángulo deseado. Por lo tanto, se ha elegido la combinación trapecio-triángulo que tarda 3,4 segundos pese a la brusquedad en los cambios de velocidad.

#### 4.2.4 Comparación entre funciones de membresía trapezoidal y triangular para el ángulo $\beta$ :

Las figuras 4.36 y 4.37 presentan las funciones de membresía para  $\beta$  en disposición trapezoidal y triangular, respectivamente:

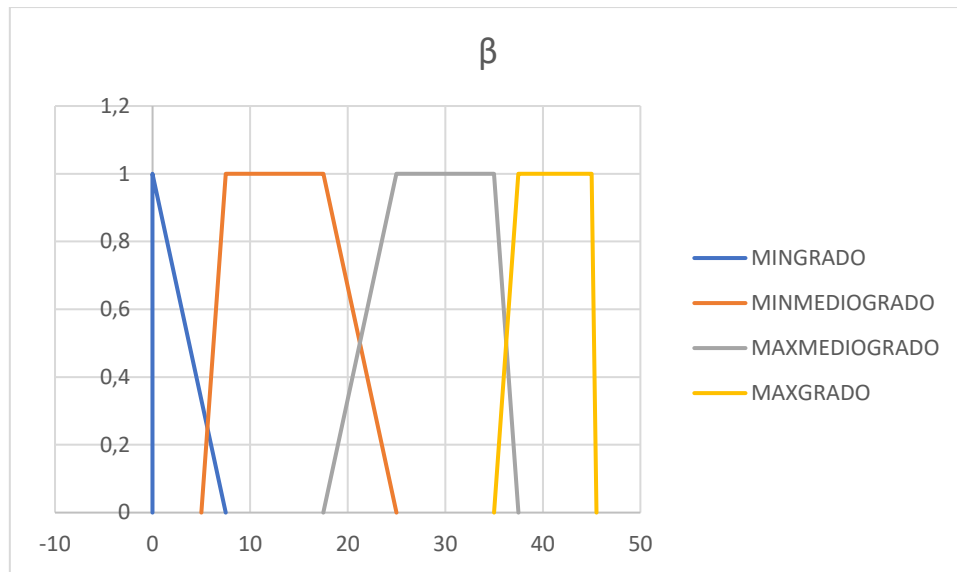


Fig. 4.36 Funciones de membresía para beta con disposición trapezoidal

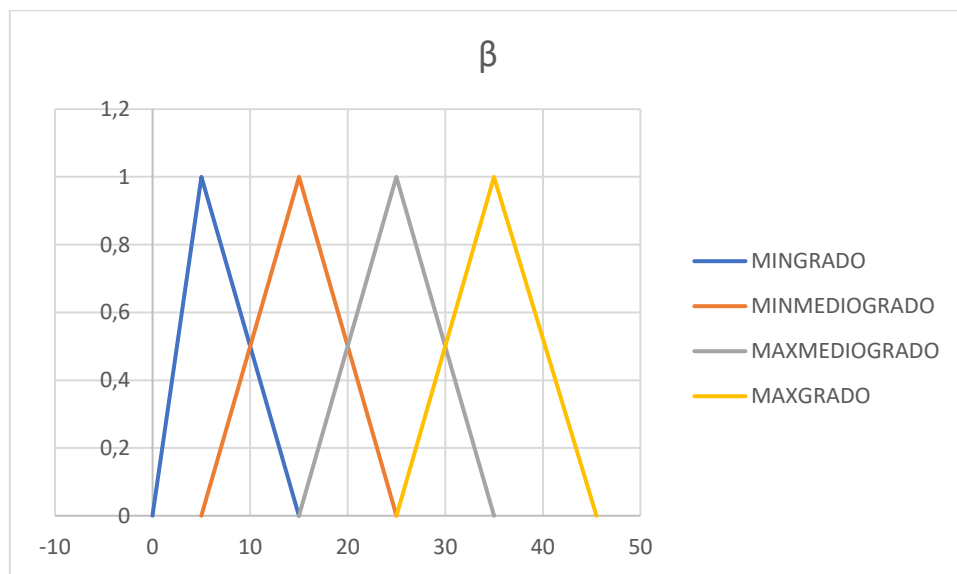


Fig. 4.37 Funciones de membresía para beta con disposición triangular

Las figuras 4.38 y 4.39 presentan las funciones de membresía para  $W_y$  en disposición triangular y trapezoidal, respectivamente:

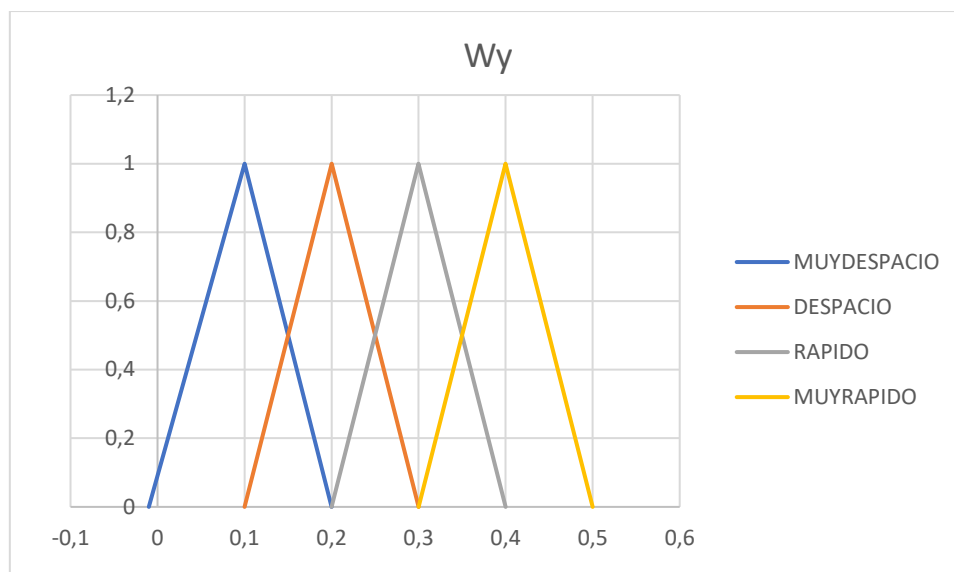


Fig. 4.38 Funciones de membresía para Wy con disposición trapezoidal

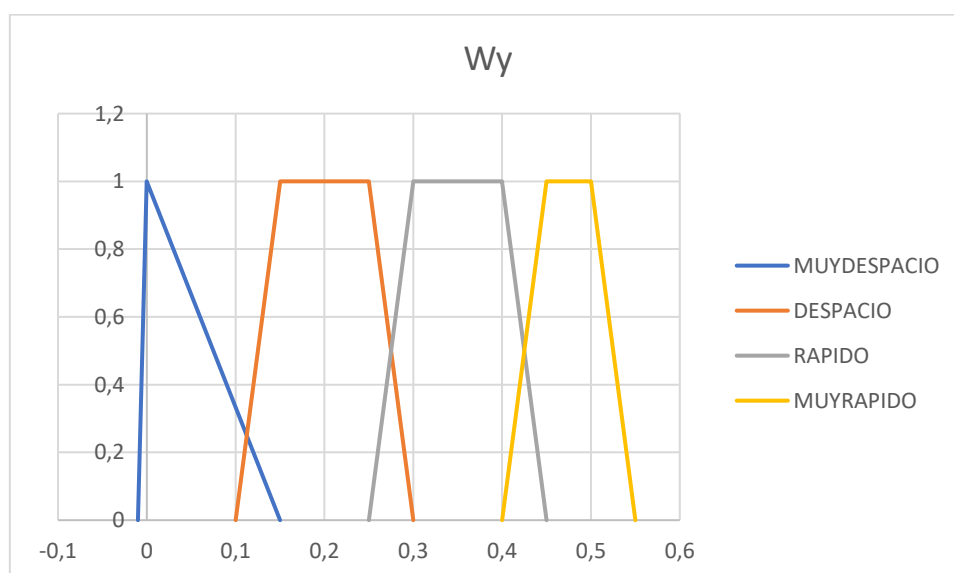


Fig. 4.39 Funciones de membresía para beta con disposición trapezoidal

La Tabla 16 presenta las siguientes reglas difusas para  $\beta$ :

Tabla 16 Reglas difusas beta

SI	ES	ENTONCES	ES
$\beta$	MINGRADO	Wy	MUYDESPACIO
	MINMEDIAGRADO		DESPACIO
	MAXMEDIAGRADO		RAPIDO
	MAXGRADO		MUYRAPIDO

En las Tablas 17-20 y en las figuras 4.40-4.47 se presentan los resultados obtenidos. Al igual que en el apartado anterior se ha decidido que una aproximación angular adecuada comienza cuando el dron se encuentra con una inclinación de  $\pm 0.5^\circ$  respecto a la plataforma objetivo:

Tabla 17 Datos beta y Wy combinación trapecio-trapecio

	Trapecio	Trapecio
	$\beta$	Wy
0	45	0,475
0,2	39,5569	0,475
0,4	34,1138	0,35
0,6	30,1031	0,35
0,8	26,0924	0,35
1	22,0817	0,289359
1,2	18,7659	0,229568
1,4	16,1353	0,2
1,6	13,8435	0,2
1,8	11,5516	0,2
2	9,25981	0,2
2,2	6,96798	0,189643
2,4	4,79484	0,07
2,6	3,99271	0,07
2,8	3,19057	0,07
3	2,38843	0,07
3,2	1,58629	0,07
3,4	0,784149	0,07
3,6	-0,017992	0
	tiempo total	3,6

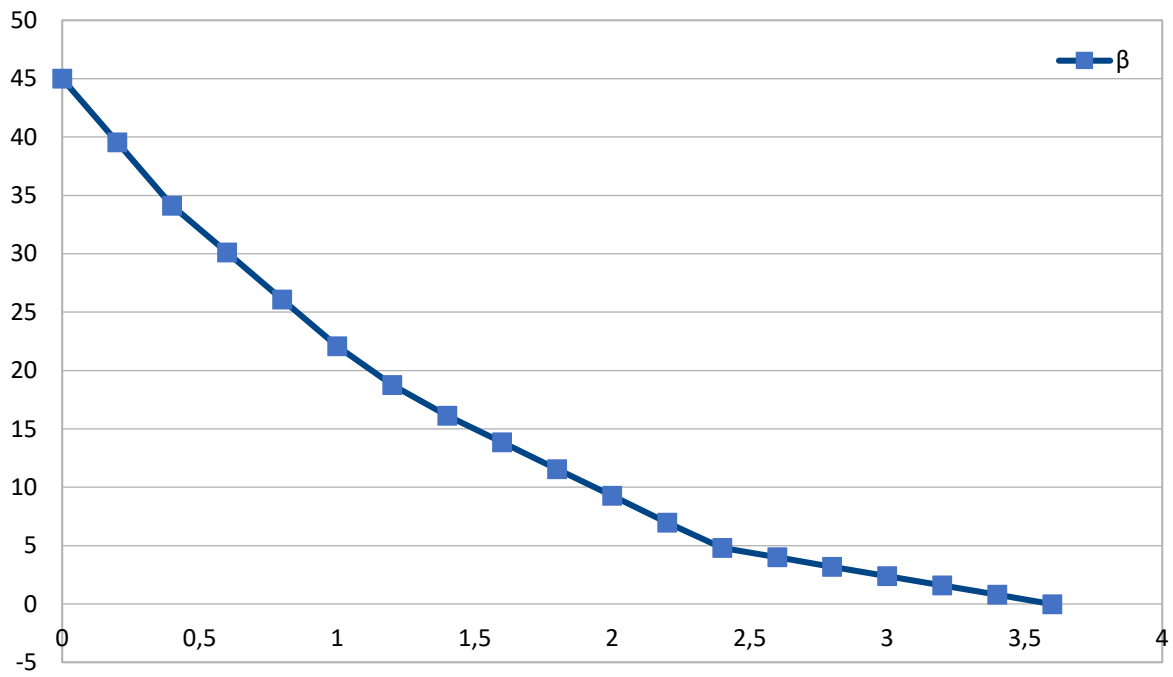


Fig. 4.40 Gráfico de la evolución de beta con combinación trapecio-trapecio

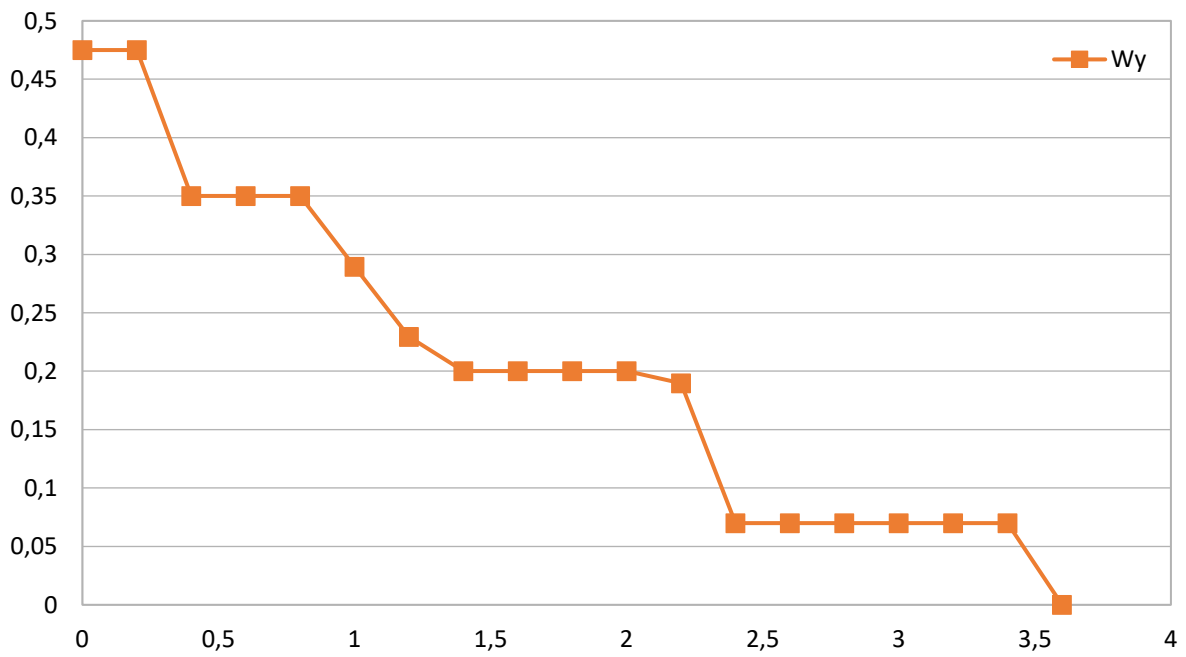


Fig. 4.41 Gráfico de la evolución de  $W_y$  con combinación trapecio-trapecio

Tabla 18 Datos beta y Wy combinación trapecio-triángulo

	Trapecio	Triángulo
	$\beta$	Wy
0	45	0,4
0,2	40,4163	0,4
0,4	35,8327	0,338443
0,6	31,9544	0,3
0,8	28,5167	0,3
1	25,079	0,3
1,2	21,6412	0,25349
1,4	18,7364	0,223722
1,6	16,1728	0,2
1,8	13,881	0,2
2	11,5891	0,2
2,2	9,29731	0,2
2,4	7,00548	0,187158
2,6	4,86082	0,095
2,8	3,7722	0,095
3	2,68358	0,095
3,2	1,59497	0,095
3,4	0,506349	0,095
3,6	-0,58227	0
	tiempo total	3,6

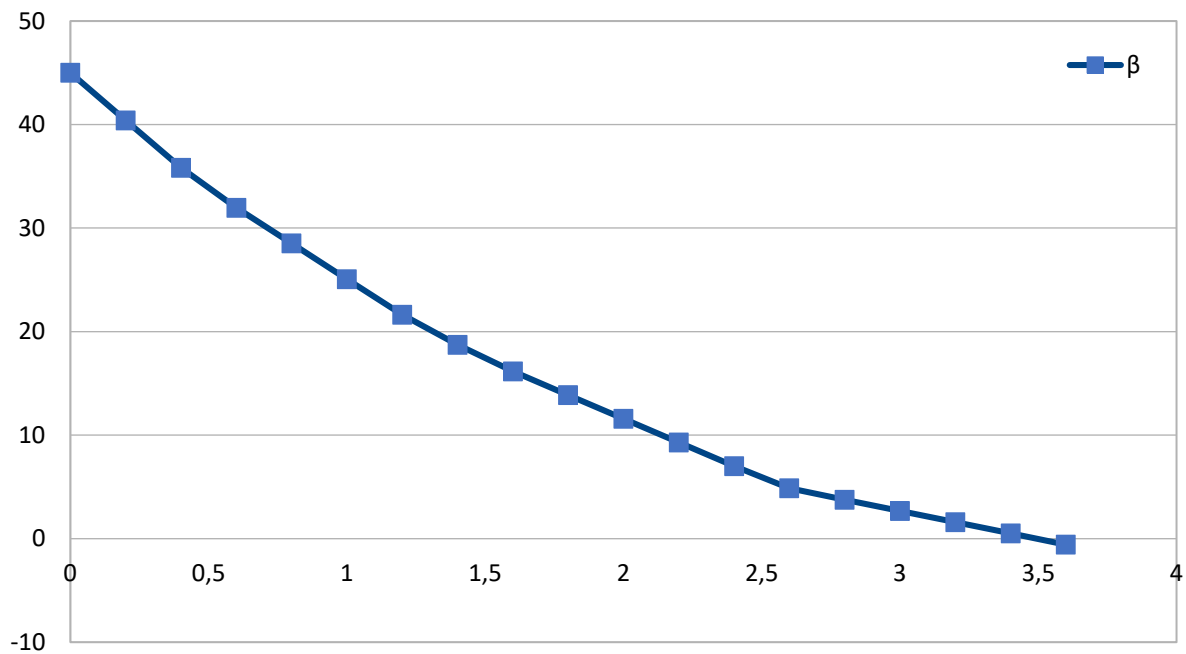


Fig. 4.42 Gráfico de la evolución de beta con combinación trapecio-triángulo



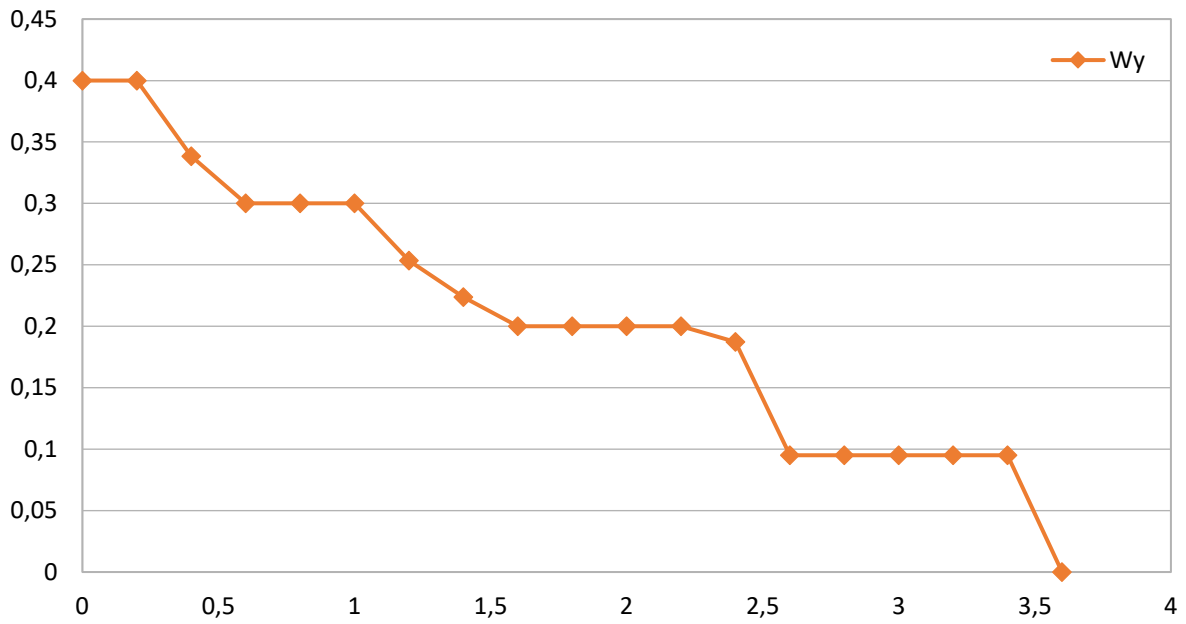


Fig. 4.43 Gráfico de la evolución de Wy con combinación trapecio-triángulo

Tabla 19 Datos beta y Wy combinación triángulo-triángulo

	Triángulo $\beta$	Triángulo Wy
0	45	0,4
0,2	40,4163	0,4
0,4	35,8327	0,4
0,6	31,249	0,358504
0,8	27,1409	0,328608
1	23,3753	0,276533
1,2	20,2065	0,251378
1,4	17,326	0,230294
1,6	14,687	0,193613
1,8	12,4683	0,16518
2	10,5755	0,150271
2,2	8,85355	0,138087
2,4	7,27119	0,125232
2,6	5,83614	0,108987
2,8	4,58725	0,095
3	3,49863	0,095
3,2	2,41001	0,095
3,4	1,3214	0,095
3,6	0,23278	0
	tiempo total	3,6

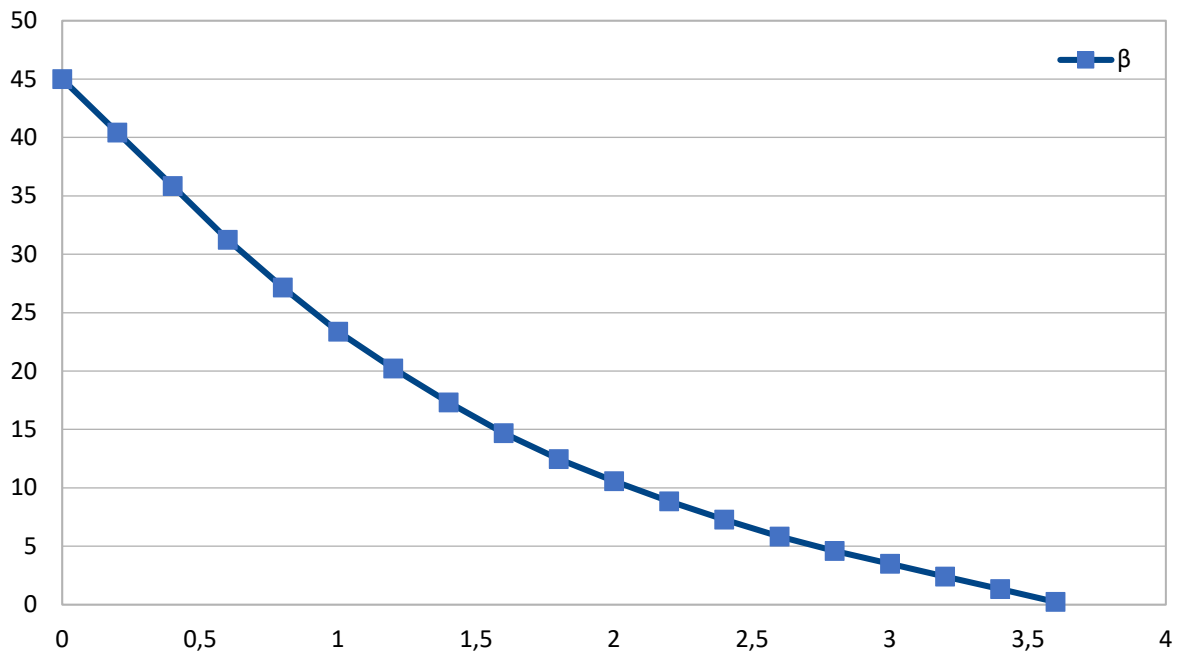


Fig. 4.44 Gráfico de la evolución de beta con combinación triángulo-triángulo

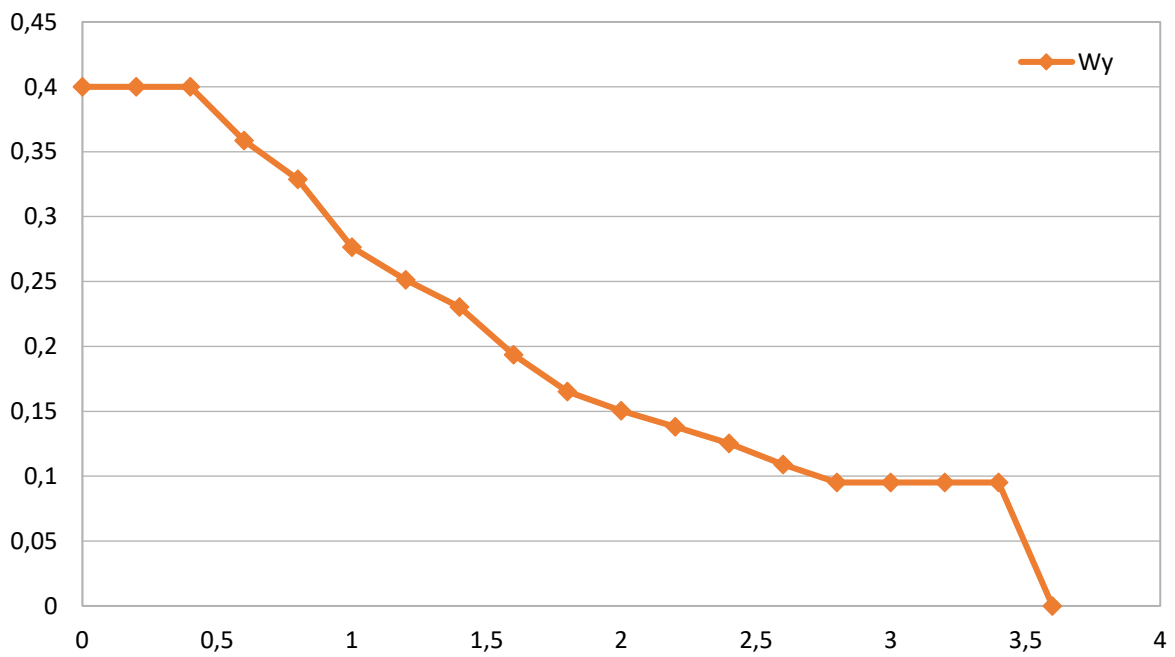


Fig. 4.45 Gráfico de la evolución de  $W_y$  con combinación triángulo-triángulo

Tabla 20 Datos beta y Wy combinación triángulo-trapecio

	Triángulo	Trapecio
	$\beta$	Wy
0	45	0,475
0,2	39,5569	0,475
0,4	34,1138	0,455745
0,6	28,8914	0,391149
0,8	24,4092	0,338774
1	20,5271	0,281788
1,2	17,2981	0,238749
1,4	14,5622	0,194155
1,6	12,3374	0,169374
1,8	10,3965	0,150881
2	8,66752	0,133749
2,2	7,13487	0,114997
2,4	5,81711	0,0917973
2,6	4,76519	0,07
2,8	3,96305	0,07
3	3,16091	0,07
3,2	2,35877	0,07
3,4	1,55663	0,07
3,6	0,754493	0,07
3,8	-0,04765	0
	tiempo total	3,8

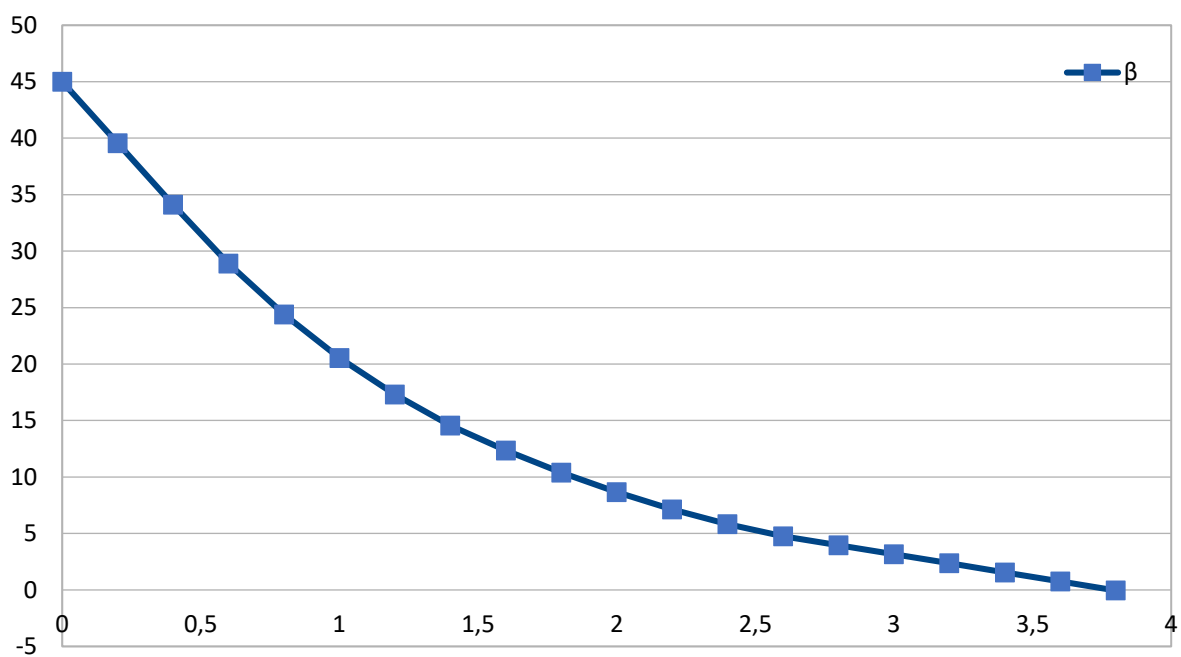


Fig. 4.46 Gráfico de la evolución de beta con combinación triángulo-trapecio

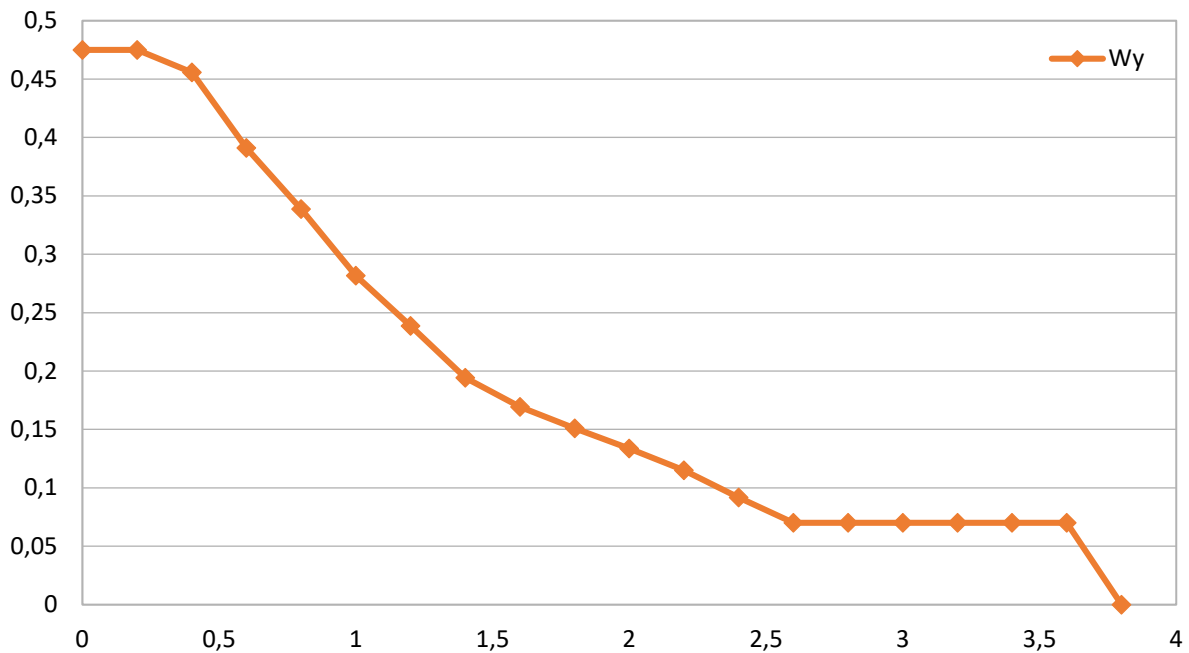


Fig. 4.47 Gráfico de la evolución de Wy con combinación triángulo-trapecio

De los resultados presentados, se puede apreciar que no existen diferencias significativas respecto al tiempo de llegada al ángulo deseado con las diferentes combinaciones. Por lo tanto, se ha decidido elegir la combinación triángulo-triángulo por tener una curva de velocidades con variaciones más uniformes que las demás.

#### 4.2.5 Análisis de la respuesta de los parámetros X, Y y Z sin variación angular:

A partir de resultados presentados en los apartados anteriores, se han seleccionado las combinaciones óptimas con el fin de analizar los resultados, comparando la variación de velocidades y desplazamientos entre sí cuando la superficie de aterrizaje es completamente horizontal. El requisito que se ha impuesto en esta prueba ha sido el mismo que en las que han participado los parámetros X, Y y Z, se detiene la velocidad cuando se encuentra a menos de 2cm del centro del objetivo.

La Tabla 21 recoge estos datos y las gráficas 4.48 y 4.49 comparan los desplazamientos y las velocidades entre sí, mientras que las gráficas 4.50-4.52 compara cada desplazamiento con su correspondiente velocidad.

Tabla 21 Datos X, Y y Z en ausencia de inclinación

t	X	VelX	Y	VelY	Z	VelZ	$\alpha$	Wy	$\beta$	Wx
0	4	2,25	4	2,25	4	2,25	0	0	0	0
0,2	3,55	2,25	3,55	2,25	3,55	2,25	0	0	0	0
0,4	3,1	2,25	3,1	2,25	3,1	2,25	0	0	0	0
0,6	2,65	1,84254	2,65	1,93318	2,65	1,84254	0	0	0	0
0,8	2,28149	1,5	2,26336	1,68863	2,28149	1,5	0	0	0	0
1	1,98149	1,46609	1,92564	1,25	1,98149	1,46609	0	0	0	0
1,2	1,68827	1,208	1,67564	1,25	1,68827	1,208	0	0	0	0
1,4	1,44668	1	1,42564	1,25	1,44668	1	0	0	0	0
1,6	1,24668	1	1,17564	1,25	1,24668	1	0	0	0	0
1,8	1,04668	1	0,925638	1,08776	1,04668	1	0	0	0	0
2	0,846676	0,878839	0,708086	0,774242	0,846676	0,878839	0	0	0	0
2,2	0,670908	0,750068	0,553238	0,6172	0,670908	0,750068	0	0	0	0
2,4	0,520894	0,545532	0,429798	0,504836	0,520894	0,545532	0	0	0	0
2,6	0,411788	0,495	0,328831	0,412438	0,411788	0,495	0	0	0	0
2,8	0,312788	0,495	0,246343	0,331615	0,312788	0,495	0	0	0	0
3	0,213788	0,495	0,18002	0,259766	0,213788	0,495	0	0	0	0
3,2	0,114788	0,495	0,128067	0,196895	0,114788	0,495	0	0	0	0
3,4	0,0157878	0	0,088688	0,143925	0,0157878	0	0	0	0	0
3,6	0,0157878	0	0,0599029	0,101501	0,0157878	0	0	0	0	0
3,8	0,0157878	0	0,0396027	0,0693199	0,0157878	0	0	0	0	0
4	0,0157878	0	0,0257387	0,0461115	0,0157878	0	0	0	0	0
4,2	0,0157878	0	0,0165164	0	0,0157878	0	0	0	0	0
tiempo total										4,2

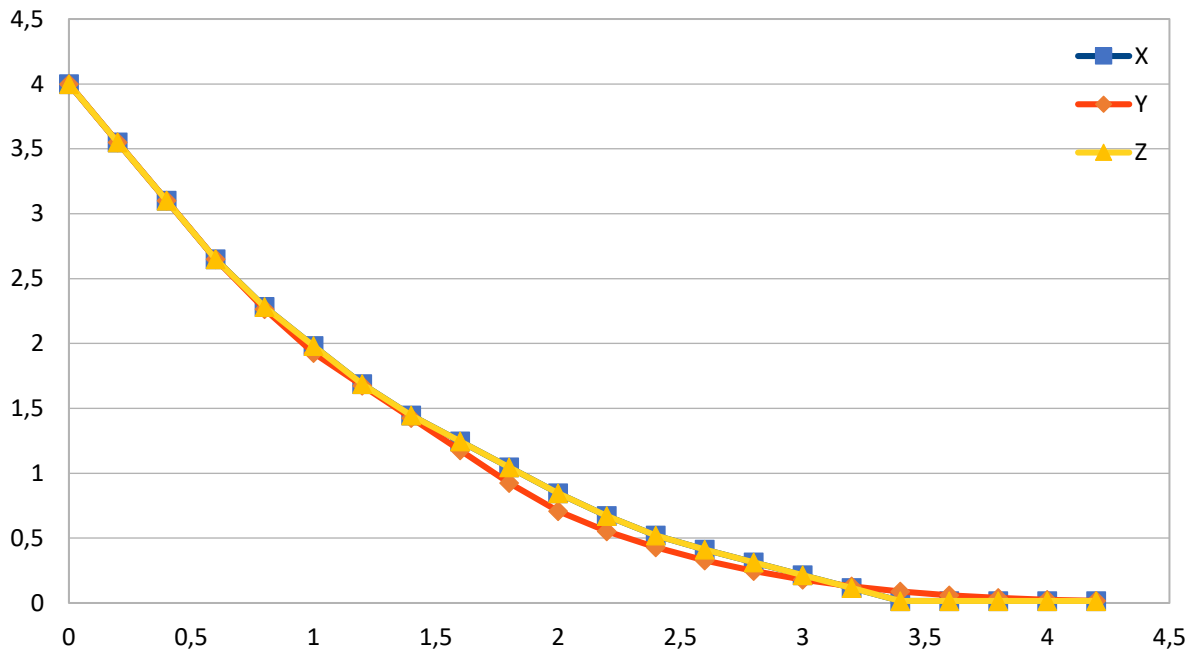


Fig. 4.48 Gráfico comparativo del desplazamiento en ausencia de inclinación

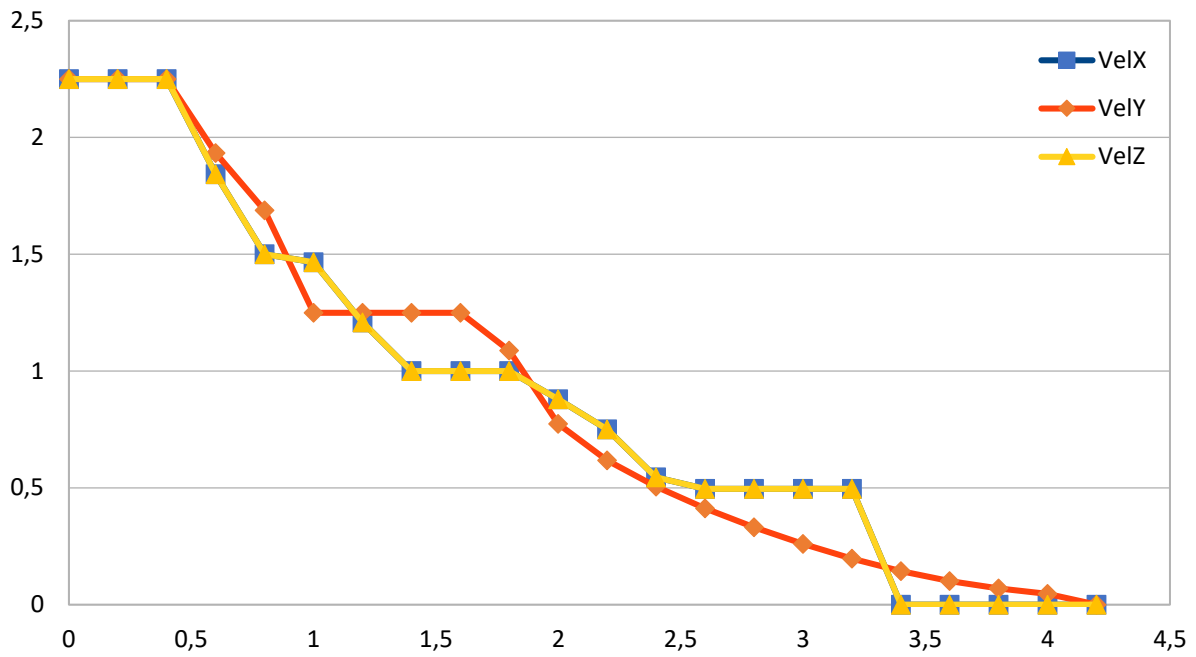


Fig. 4.49 Gráfico comparativo de la velocidad en ausencia de inclinación

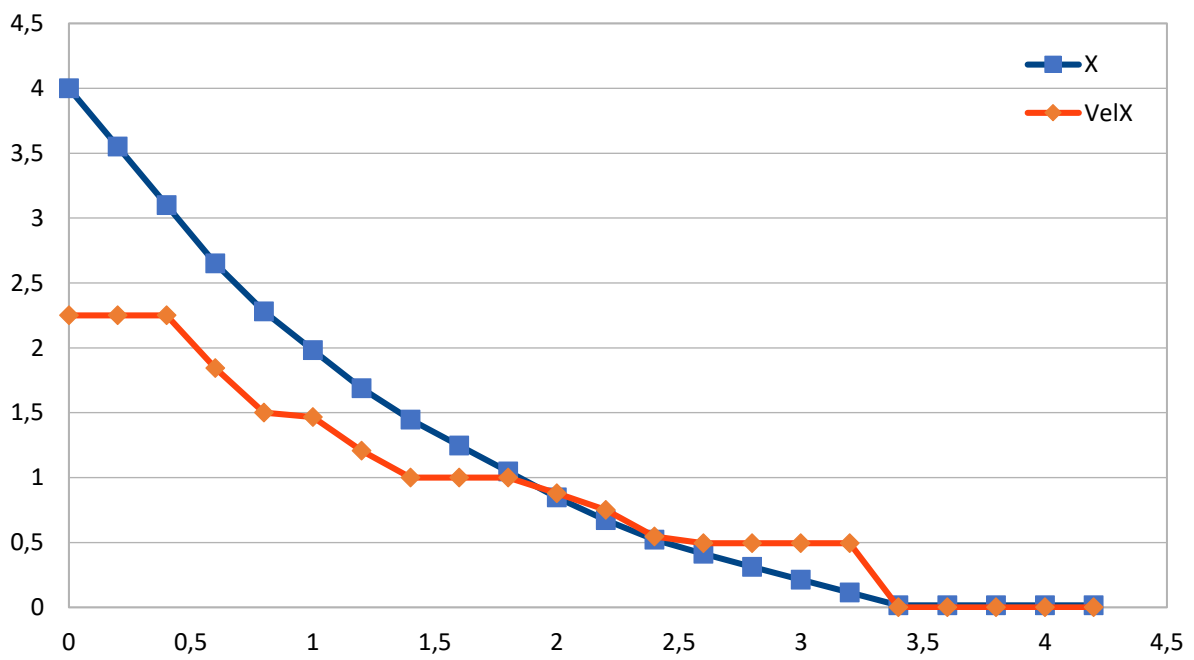


Fig. 4.50 Gráfico comparativo X y VelX en ausencia de inclinación

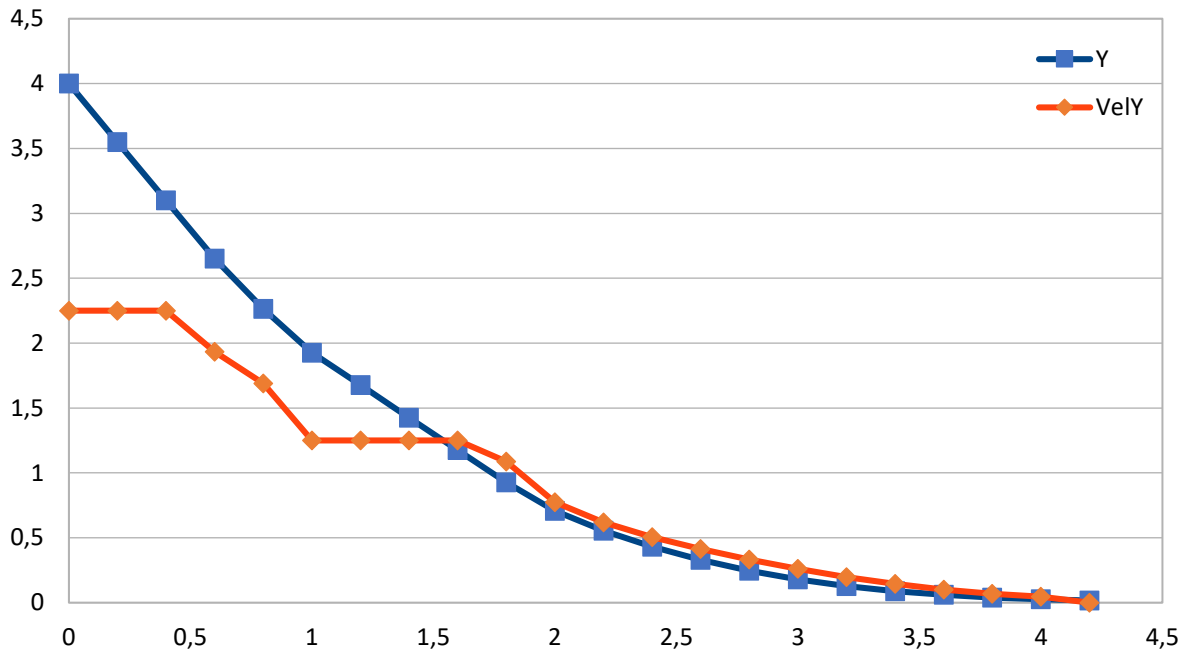


Fig. 4.51 Gráfico comparativo Y y VelY en ausencia de inclinación

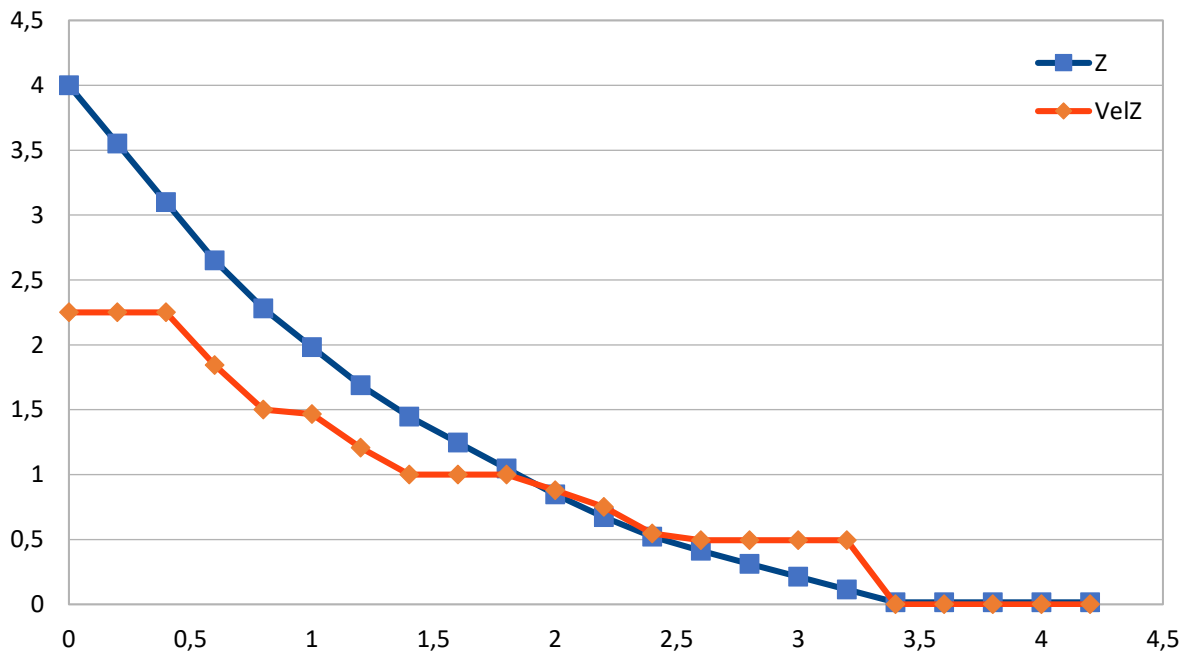


Fig. 4.52 Gráfico comparativo de Z y VelZ en ausencia de inclinación

Como se podía prever en las pruebas anteriores pruebas, en ausencia de inclinación los parámetros X y Z se comportan igual y llegan antes que el parámetro Y. La curva de los desplazamientos en ambos casos es muy similar mientras que la curva de velocidades se presenta con cambios bruscos para Z y X pero con un decrecimiento progresivo para Y.

#### 4.2.6 Análisis de la respuesta de los parámetros $X$ , $Y$ y $Z$ en presencia de variación angular $\alpha$ :

En este análisis y en los sucesivos en los que haya variación angular se han tenido que hacer algunas modificaciones respecto a los experimentos previos. Debido a la constante variación de los ejes se han presentado casos en los que el dron excedía la posición de parada adentrándose en valores negativos en los ejes  $X$  y  $Z$  con lo que se ha aumentado a 6cm la distancia y ha sido necesario implementar una nueva región en las funciones de membresía llamada ‘DEMASIADOCERCA’ con el fin de rectificar dicho exceso.

Las figuras 4.53 y 4.54 presentan las nuevas funciones de membresía para  $Z$  y  $VelZ$  con una región de rectificación para contrarrestar las posiciones negativas en las que se pueda encontrar la aeronave:

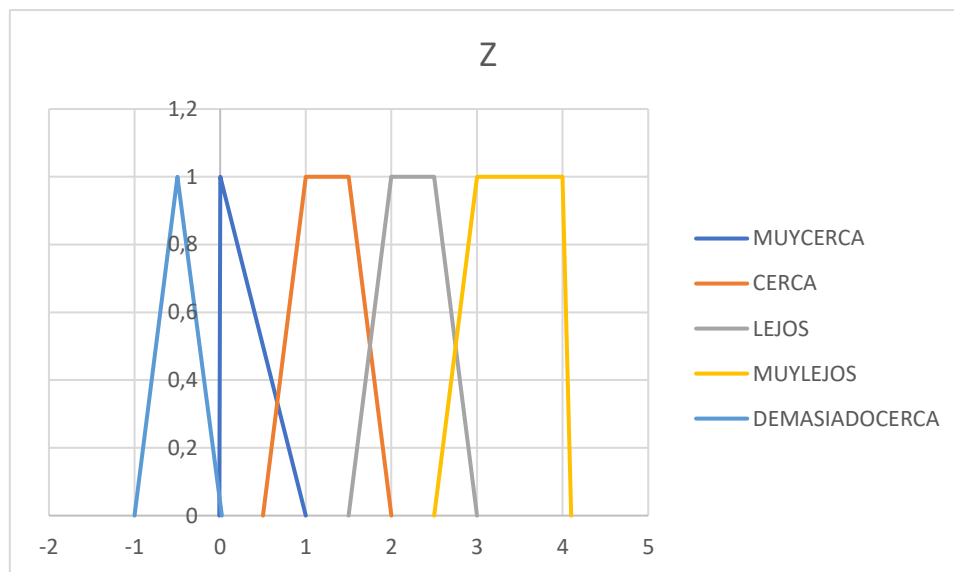


Fig. 4.53 Nuevas funciones de membresía para  $Z$  con región de rectificación



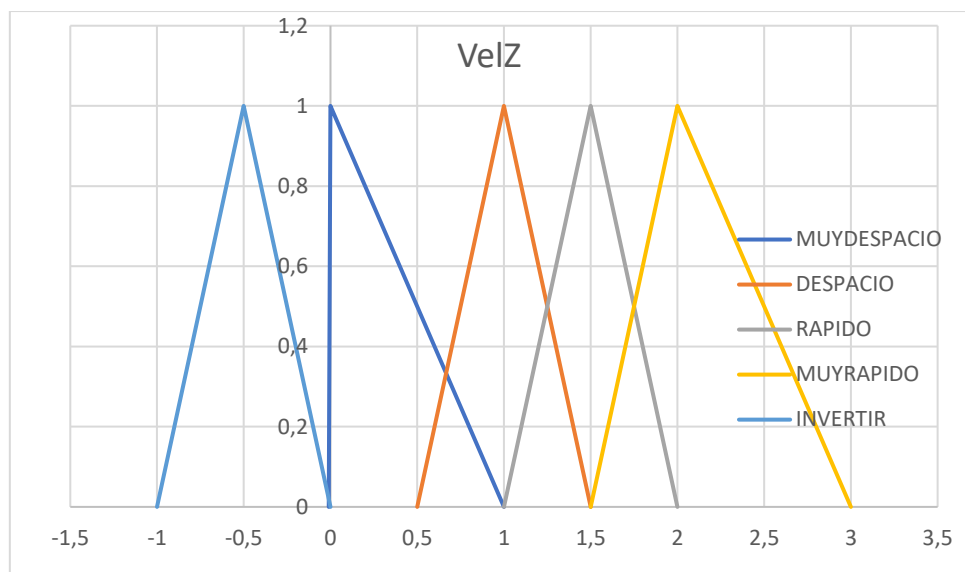


Fig. 4.54 Nuevas funciones de membresía para VelZ con región de rectificación

La Tabla 22 muestra los resultados que se han obtenido en presencia de la inclinación en  $\alpha$  son:

Tabla 22 Datos X, Y, Z y alfa en ausencia de beta

tiempo	x	VelX	y	VelY	z	VelZ	alfa	Wx	beta	Wx
0	4	2,25	4	2,25	4	2,25	-45	-0,3	0	0
0,2	3,55416	2,20685	3,54591	2,10849	3,54992	2,42314	-41,5623	-0,3	0	0
0,4	3,12031	2,21566	3,11706	2,01491	3,06493	2,08485	-38,1245	-0,3	0	0
0,6	2,68721	1,89874	2,70487	1,9386	2,64723	1,91435	-34,6868	-0,3	0	0
0,8	2,31913	1,47804	2,30661	1,67879	2,2632	1,61541	-31,249	-0,3	0	0
1	2,0362	1,48073	1,95935	1,44468	1,9385	1,52329	-27,8113	-0,3	0	0
1,2	1,75319	1,24613	1,65862	1,09217	1,63182	1,42254	-24,3736	-0,289139	0	0
1,4	1,51722	1,03772	1,42844	0,850351	1,34495	1,16518	-21,0603	-0,257176	0	0
1,6	1,32272	0,990415	1,24697	0,882472	1,10931	1,02364	-18,1133	-0,236795	0	0
1,8	1,13687	0,990415	1,05978	0,833022	0,901871	1,06427	-15,3998	-0,207278	0	0
2	0,949709	0,946842	0,883614	0,807875	0,68638	1,06636	-13,0246	-0,2	0	0
2,2	0,769576	0,84519	0,713808	1,35274	0,470657	-0,26918	-10,7328	-0,2	0	0
2,4	0,607472	0,678725	0,436369	0,278845	0,522283	0,8392	-8,44093	-0,2	0	0
2,6	0,477715	0,494103	0,374903	0,419201	0,35249	0,404338	-6,1491	-0,2	0	0
2,8	0,383773	0,495737	0,286264	0,377674	0,269868	0,284678	-3,85728	-0,140138	0	0
3	0,28839	0,495624	0,207008	0,298673	0,211512	0,237754	-2,25142	-0,0933493	0	0
3,2	0,192051	0,495329	0,144745	0,225972	0,16297	0,199362	-1,18172	-0,0612547	0	0
3,4	0,0949727	0,357113	0,0982829	0,168192	0,122598	0,159281	-0,479794	-0,0311392	0	0
3,6	0,0249164	0,240517	0,0643081	0,122542	0,0906159	0,121531	-0,122966	-0,0092715	0	0
3,8	-0,0222813	-0,499867	0,0400966	0,0964723	0,0664412	0,0798307	-0,0167227	-0,0013267	0	0
4	0,0782442	0,324451	0,0197483	0,0550202	0,0500438	0,0652244	-0,0015195	-0,0001215	0	0
4,2	0,0140544	0	0,0321612	0	0,0197949	0	-0,0001276	0	0	0

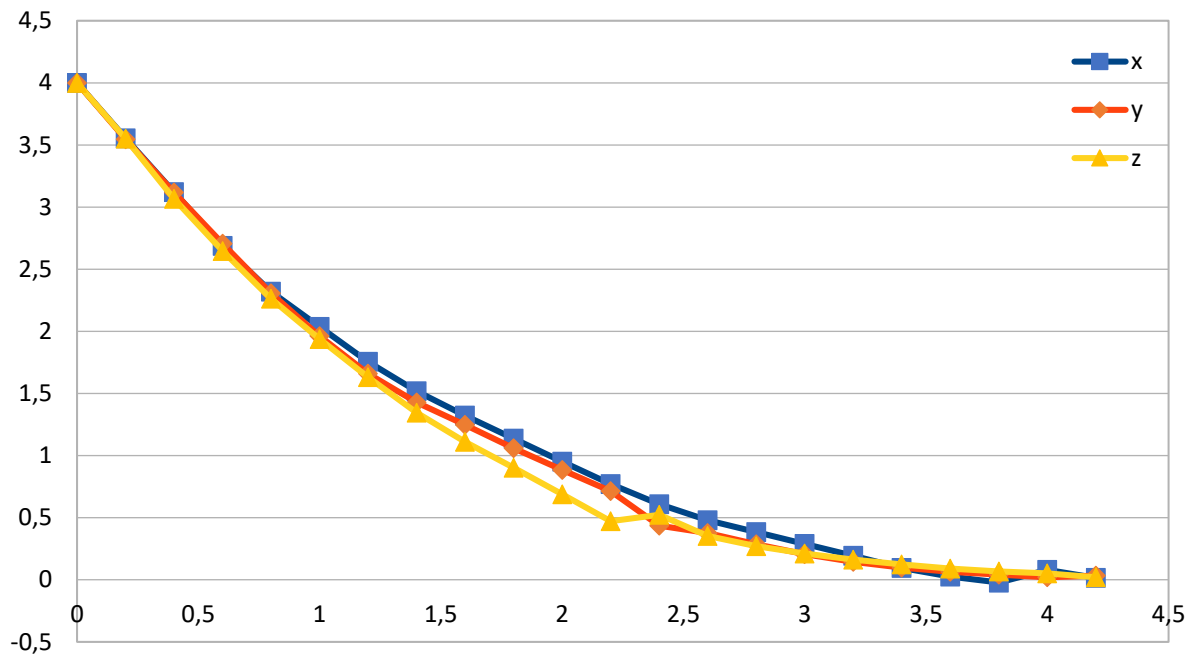


Fig. 4.55 Gráfico comparativo del desplazamiento en presencia de ángulo alfa

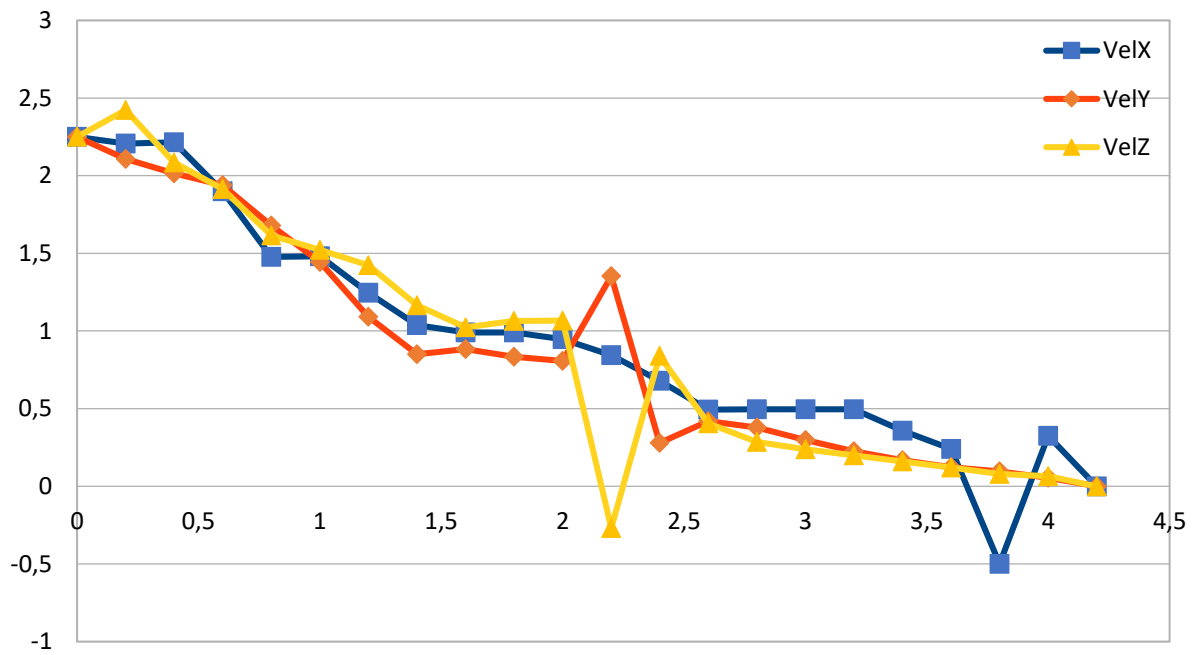


Fig. 4.56 Gráfico comparativo para las velocidades en presencia de ángulo alfa

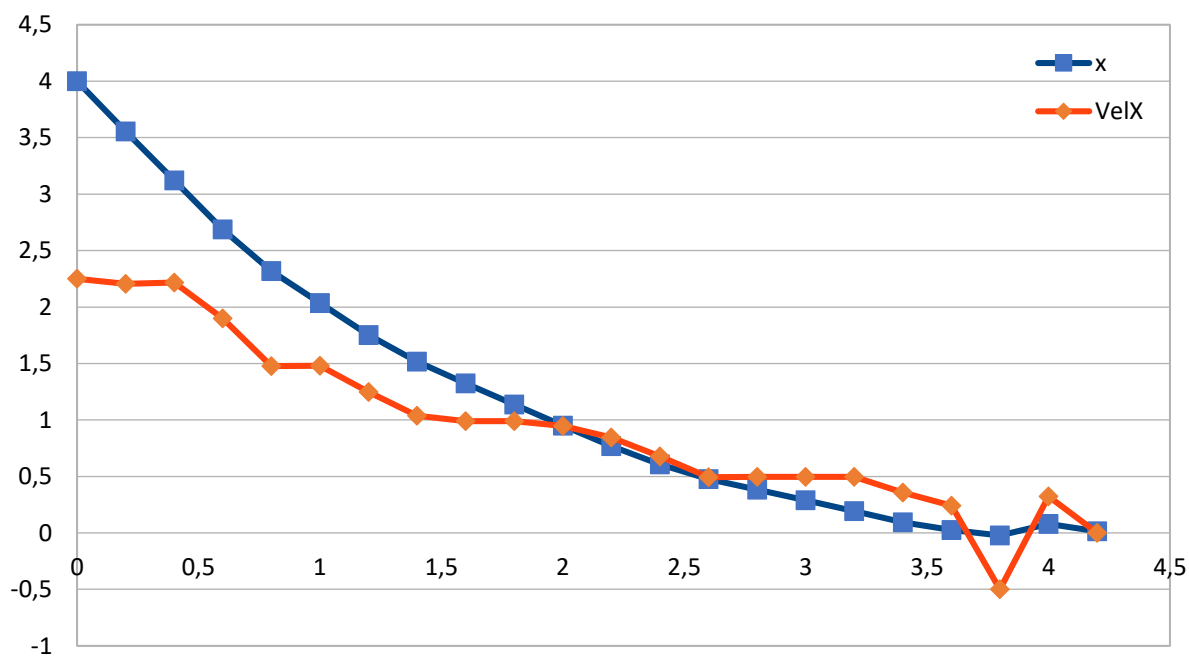


Fig. 4.57 Gráfico comparativo de X y VelX en presencia de alfa

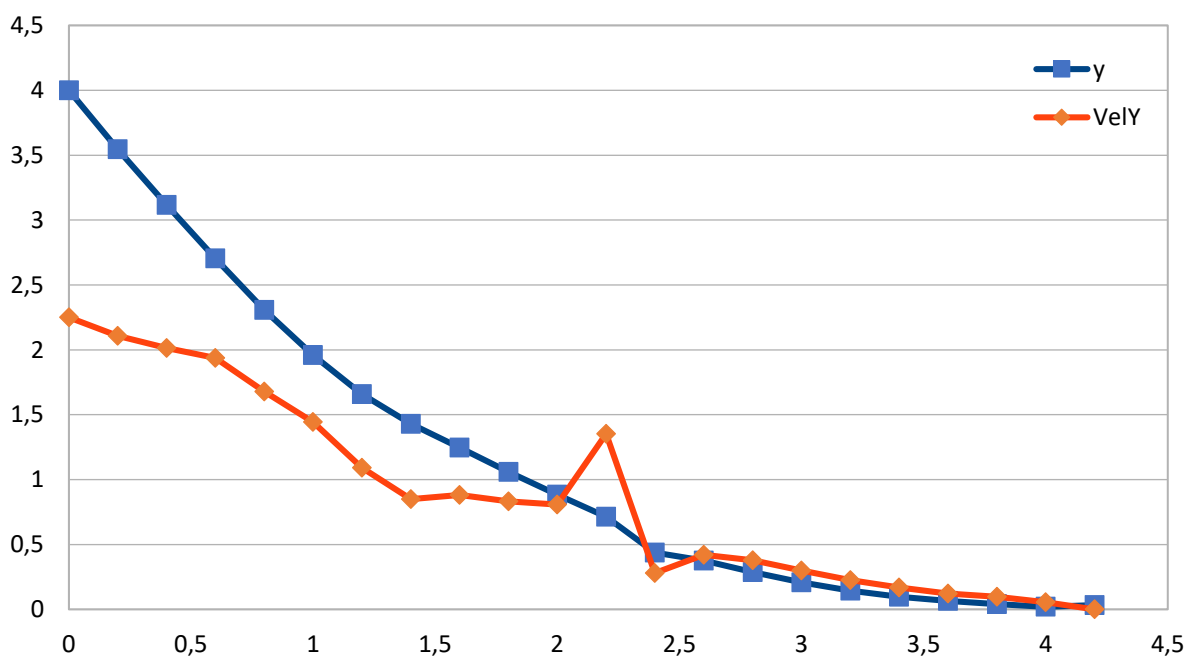


Fig. 4.58 Gráfico comparativo de Y y VelY en presencia de alfa

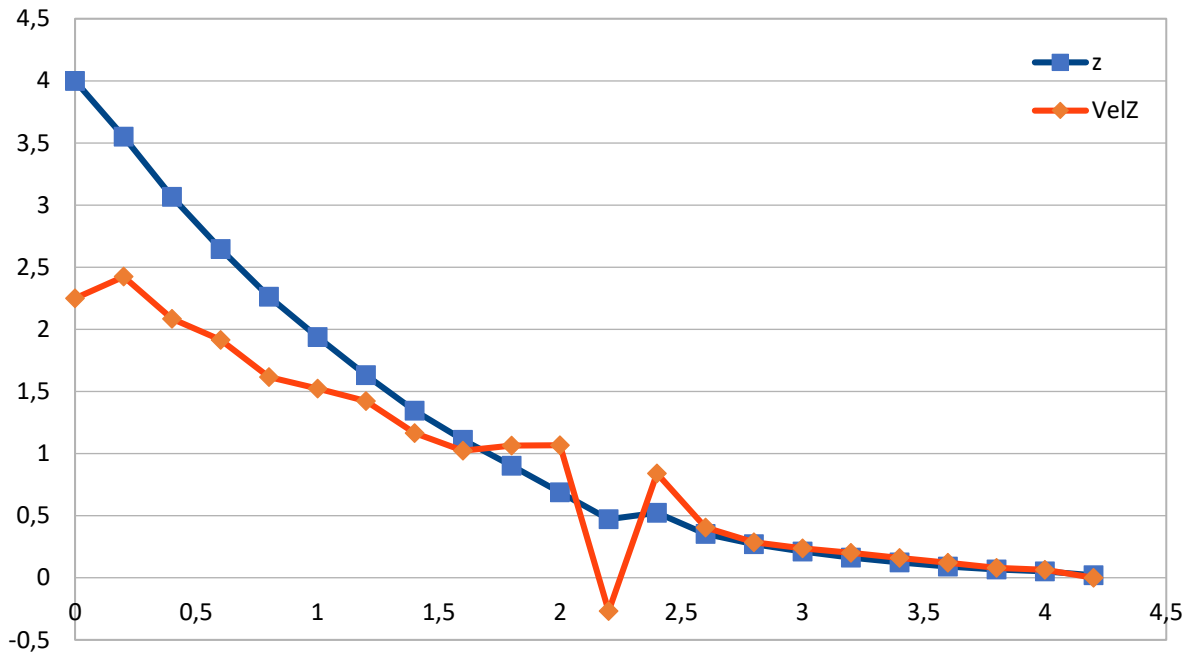


Fig. 4.59 Gráfico comparativo entre Z y VelZ en presencia de alfa

De los resultados presentados se deduce que, en este experimento, aunque el acercamiento es gradual y progresivo, la incorporación de esta nueva región difusa de rectificación y el hecho de que roten los ejes obliga a la velocidad en los tres parámetros a cambiar puntualmente de forma muy brusca de valor hasta llegar a la posición deseada. Pese a esto, el desplazamiento no sufre variaciones indeseables, la variación de velocidades, en general es suave y adecuada a la aproximación, la posición final del dron se encuentra a  $X=1\text{cm}$ ,  $Y=3\text{cm}$  y  $Z=1\text{cm}$ , con inexistencia de valores negativos en el eje Z y un tiempo de aproximación de 4,2s; por lo tanto, se puede determinar que el aterrizaje ha sido satisfactorio.

#### 4.2.7 Análisis de la respuesta de los parámetros X, Y y Z en presencia de variación angular $\beta$ :

La Tabla 23 muestra los resultados obtenidos en presencia de  $\beta$ :

Tabla 23 Datos X, Y, Z, beta en ausencia de alfa

tiempo	x	VelX	y	VelY	z	VelZ	alfa	Wx	beta	Wy
0	4	2,25	4	2,25	4	2,25	0	0	45	0,4
0,2	3,55	2,063	3,55	2,25	3,55	2,42261	0	0	40,4163	0,4
0,4	3,1374	1,96624	3,1	2,25	3,06548	1,93872	0	0	35,8327	0,4
0,6	2,74415	1,8485	2,65	1,93318	2,67773	1,91198	0	0	31,249	0,358504
0,8	2,37445	1,77409	2,26336	1,68863	2,29534	1,62225	0	0	27,1409	0,328608
1	2,01963	1,28277	1,92564	1,25	1,97089	1,58424	0	0	23,3753	0,276533
1,2	1,76308	1,01725	1,67564	1,25	1,65404	1,37477	0	0	20,2065	0,251378
1,4	1,555963	1,09851	1,42564	1,25	1,37908	1,13504	0	0	17,3259	0,230294
1,6	1,33993	0,822474	1,17564	1,25	1,15208	1,05426	0	0	14,687	0,193613
1,8	1,17543	0,576899	0,925638	1,08776	0,941225	0,955098	0	0	12,4683	0,16518
2	1,06005	0,825096	0,708086	0,774242	0,750206	0,564993	0	0	10,5755	0,150271
2,2	0,895034	1,39736	0,553238	0,6172	0,637207	0,217659	0	0	8,85354	0,138087
2,4	0,615561	0,423432	0,429798	0,504836	0,680739	0,953464	0	0	7,27119	0,125232
2,6	0,530875	0,545244	0,328831	0,412438	0,490046	0,583717	0	0	5,83614	0,108987
2,8	0,421826	0,496251	0,246343	0,331615	0,373302	0,379801	0	0	4,58725	0,095
3	0,322576	0,396494	0,18002	0,259766	0,297342	0,298881	0	0	3,49863	0,095
3,2	0,243277	0,387635	0,128067	0,196895	0,237566	0,309742	0	0	2,41001	0,095
3,4	0,16575	0,376121	0,088688	0,143925	0,175617	0,322873	0	0	1,32139	0,095
3,6	0,0905256	0,347378	0,0599029	0,101501	0,111043	0,320543	0	0	0,232777	0,095
3,8	0,0483402	0	0,0396027	0	0,0175827	0	0	0	-0,85584	0

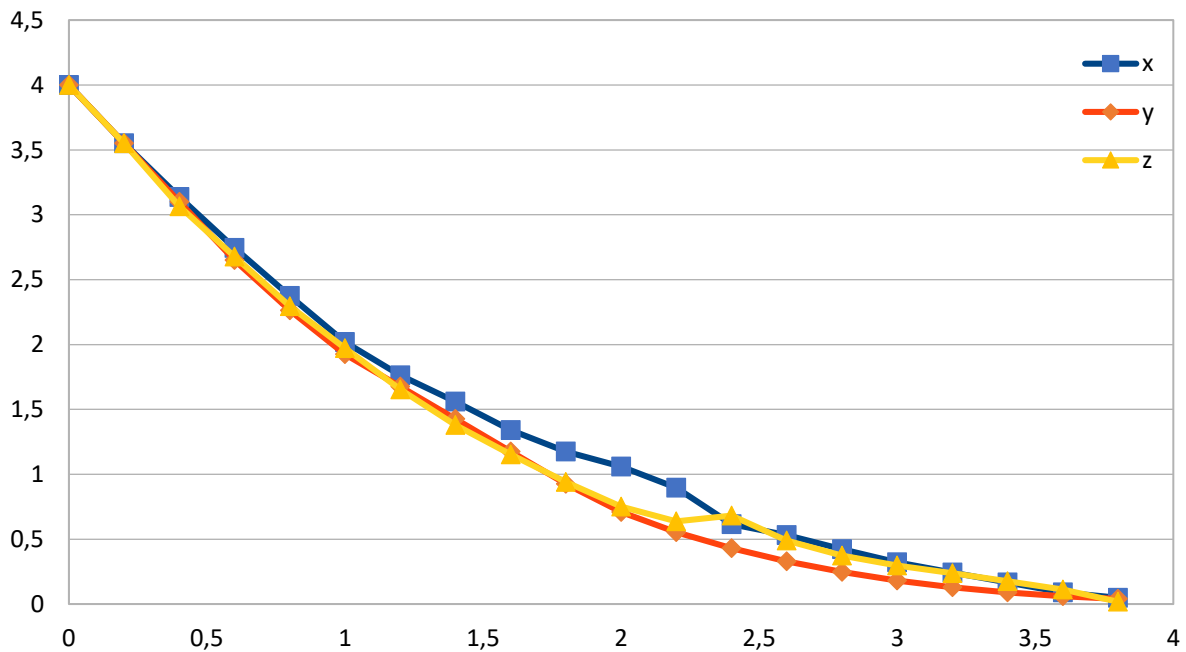


Fig. 4.60 Gráfico comparativo del desplazamiento en presencia de beta

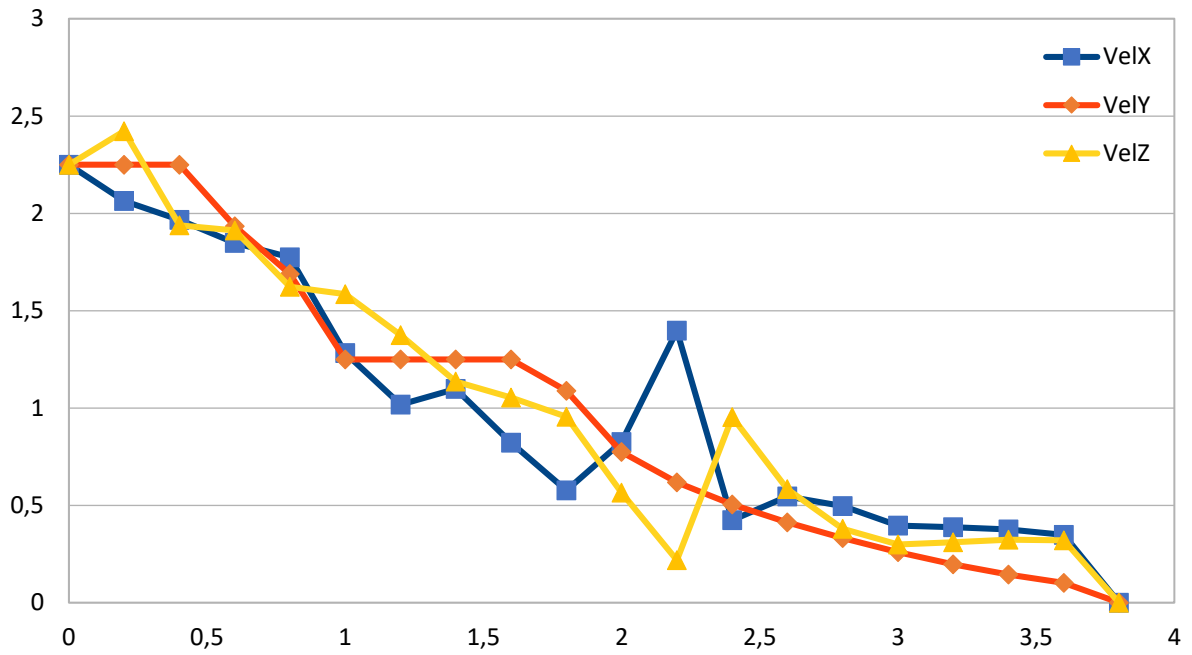


Fig. 4.61 Gráfico comparativo de las velocidades en presencia de beta

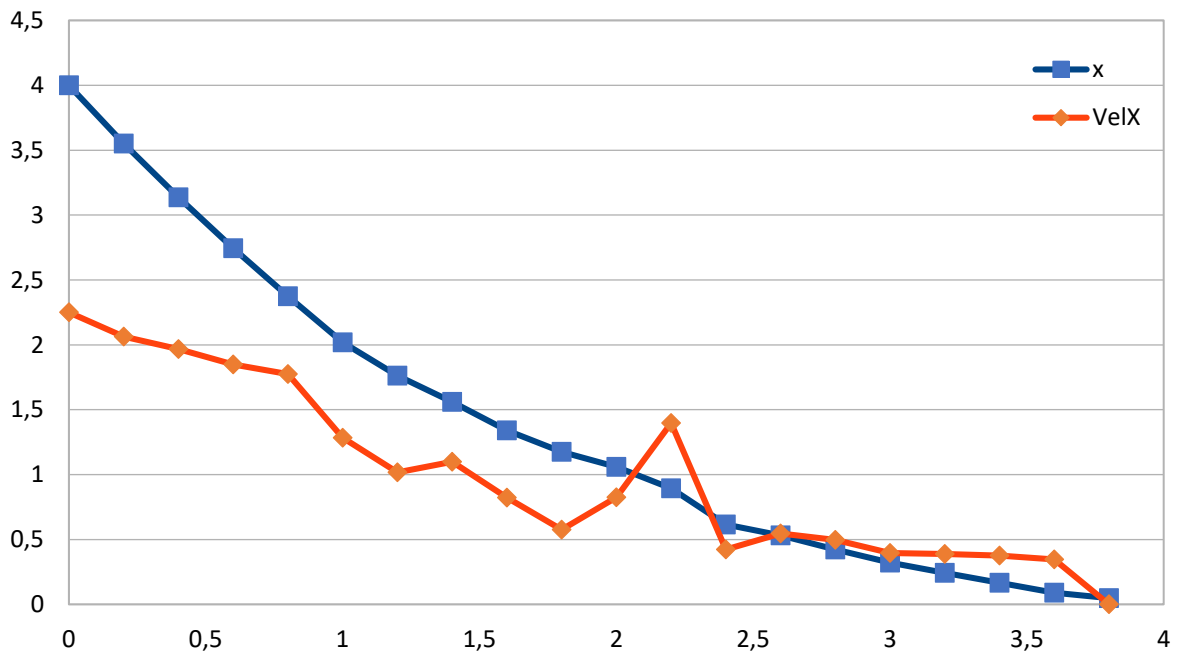


Fig. 4.62 Gráfico comparativo de X y VelX en presencia de beta

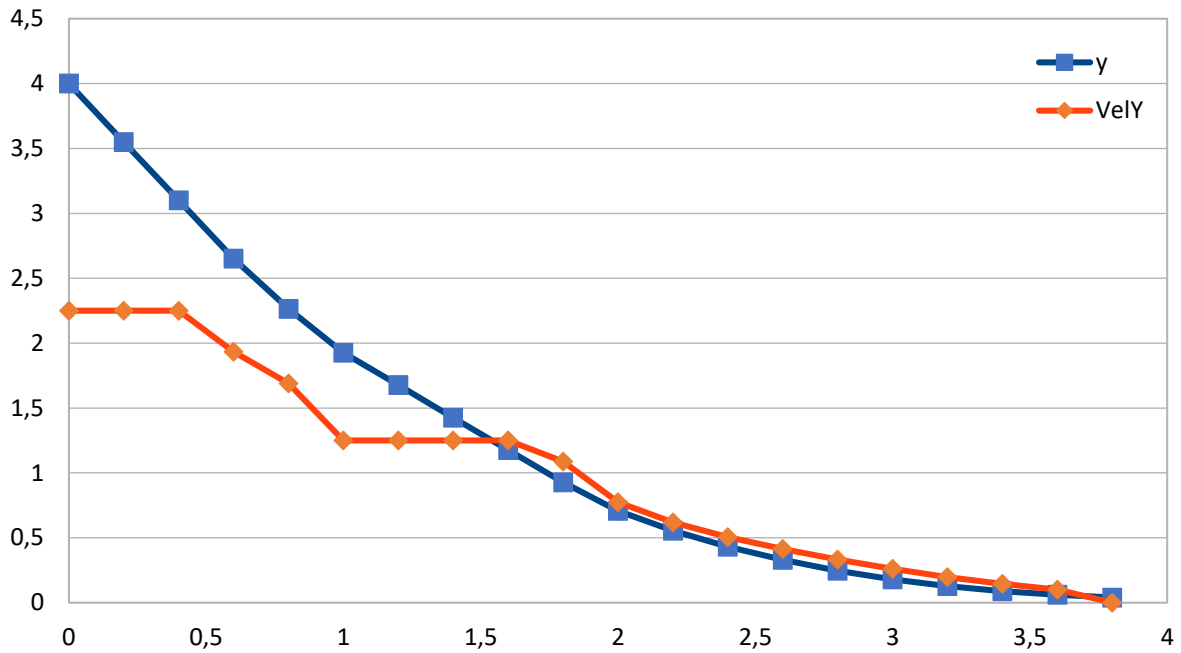


Fig. 4.63 Gráfico comparativo de Y y VelY en presencia de beta

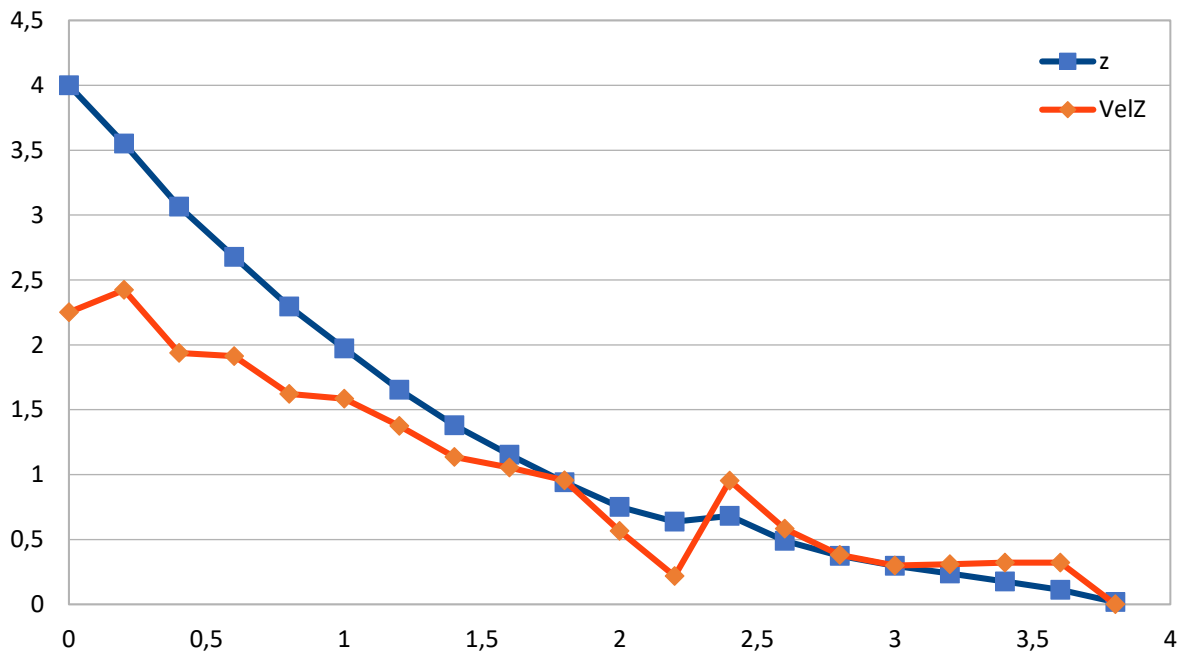


Fig. 4.64 Gráfico comparativo de Z y VelZ en presencia de beta

Como en la anterior simulación, en esta se presentan desplazamientos parecidos entre los tres ejes y grandes variaciones puntuales en las velocidades, excepto en el eje Y. De ello deriva que las funciones de membresía seleccionadas para el eje Y generan una respuesta más robusta que la que se generan en los otros dos ejes. Además, cabe destacar, que una variación en la inclinación  $\beta$  impacta menos en el tiempo de llegada.

Como en el escenario anterior el aterrizaje se ha completado satisfactoriamente, situándose el dron en las coordenadas X=4, Y=3 y Z=1 con un tiempo de 3,8s y sin valores negativos en la Z.

#### 4.2.8 Análisis de la respuesta de los parámetros X, Y y Z en presencia de variación angular $\alpha$ y $\beta$ :

En esta última prueba se ha probado el controlador en condiciones máximas de operación, esto es, 4m para todas las coordenadas cartesianas,  $-45^\circ$  para  $\alpha$  y  $45^\circ$  para  $\beta$ , en la Tabla 24, se adjuntan los resultados del test:

Tabla 24 Datos X, Y, Z, alfa y beta

tiempo	x	VelX	y	VelY	z	VelZ	alfa	Wx	beta	Wy
0	4	2,25	4	2,25	4	2,25	-45	-0,3	45	0,4
0,2	3,56748	2,063	3,53227	2,10068	3,55017	2,55317	-41,5623	-0,3	40,4163	0,4
0,4	3,2171	1,98228	3,04671	2,01363	3,04059	2,09543	-38,1245	-0,3	35,8327	0,4
0,6	2,94036	1,94781	2,50792	1,86082	2,62366	1,86951	-34,6868	-0,3	31,249	0,358504
0,8	2,71844	1,87912	1,91777	1,38911	2,24981	1,88895	-31,249	-0,3	27,1409	0,328608
1	2,53565	1,73648	1,33499	0,83338	1,8628	1,53574	-27,8113	-0,3	23,3753	0,276533
1,2	2,37424	1,67646	0,780133	0,912561	1,52587	1,12395	-24,3736	-0,289139	20,2065	0,251378
1,4	2,15978	1,79285	0,14089	0,708159	1,23402	1,07676	-21,0603	-0,257176	17,3259	0,230294
1,6	1,80855	1,15097	-0,487735	-0,360927	1,09806	0,941903	-18,1133	-0,236795	14,687	0,193613
1,8	1,46223	0,899399	-0,888571	-0,912537	0,750818	0,635546	-15,3998	-0,207278	12,4683	0,16518
2	1,05292	0,545038	-1,11017	-1,17456	0,436855	0,414531	-13,0246	-0,2	10,5755	0,150271
2,2	0,625485	0,493206	-1,1686	-1,37423	0,159601	0,241393	-10,7328	-0,2	8,85354	0,138087
2,4	0,172861	0,0885995	-1,0276	-1,19306	-0,0538419	-0,0202216	-8,44093	-0,2	7,27119	0,125232
2,6	-0,176512	-0,0459307	-0,769427	-0,973059	-0,160877	-0,0120463	-6,1491	-0,2	5,83614	0,108987
2,8	-0,414455	-0,7016	-0,418723	-0,422399	-0,190842	-0,00299	-3,85728	-0,140138	4,58725	0,095
3	-0,409773	-0,702476	-0,153726	-0,224805	-0,111782	-0,154138	-2,25142	-0,0933493	3,49863	0,095
3,2	-0,297131	-0,703098	0,0397112	-0,0094394	-0,0320969	0,0270479	-1,18172	-0,0612547	2,41001	0,095
3,4	-0,12032	-0,288741	0,114496	0,395956	0,00683558	-0,201949	-0,479794	-0,0311392	1,32139	0,095
3,6	-0,0291072	-0,283504	0,0470077	0,39223	0,0658402	-0,208048	-0,122966	-0,0092715	0,232777	0,095
3,8	0,0232049	0,255433	-0,0671786	-0,157009	0,0907953	0,200008	-0,0167227	-0,0013267	-0,85584	0
4	-0,0343952	0,15698	-0,0305832	-0,100583	0,0501898	0,136443	-0,0015195	-0,0001215	-0,85584	0
4,2	-0,0293875	0	0,0372607	0	0,0520627	0	-0,0001276	0	-0,85584	0



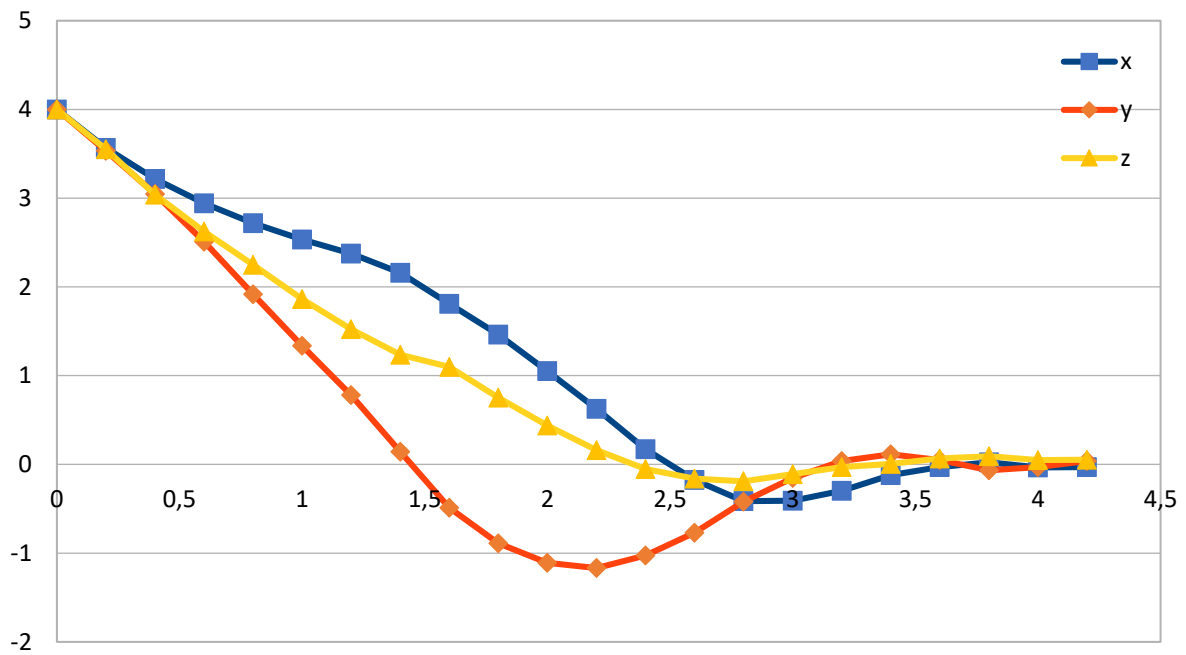


Fig. 4.65 Gráfico comparativo del desplazamiento en presencia de alfa y beta

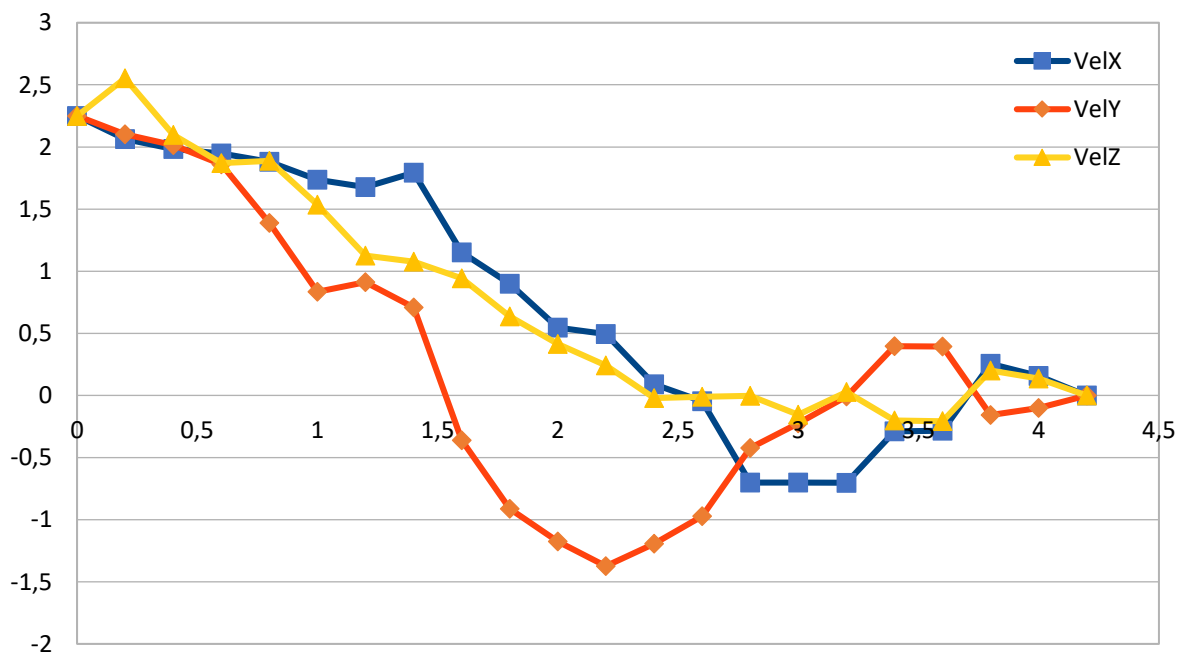


Fig. 4.66 Gráfico comparativo de las velocidades en presencia de alfa y beta

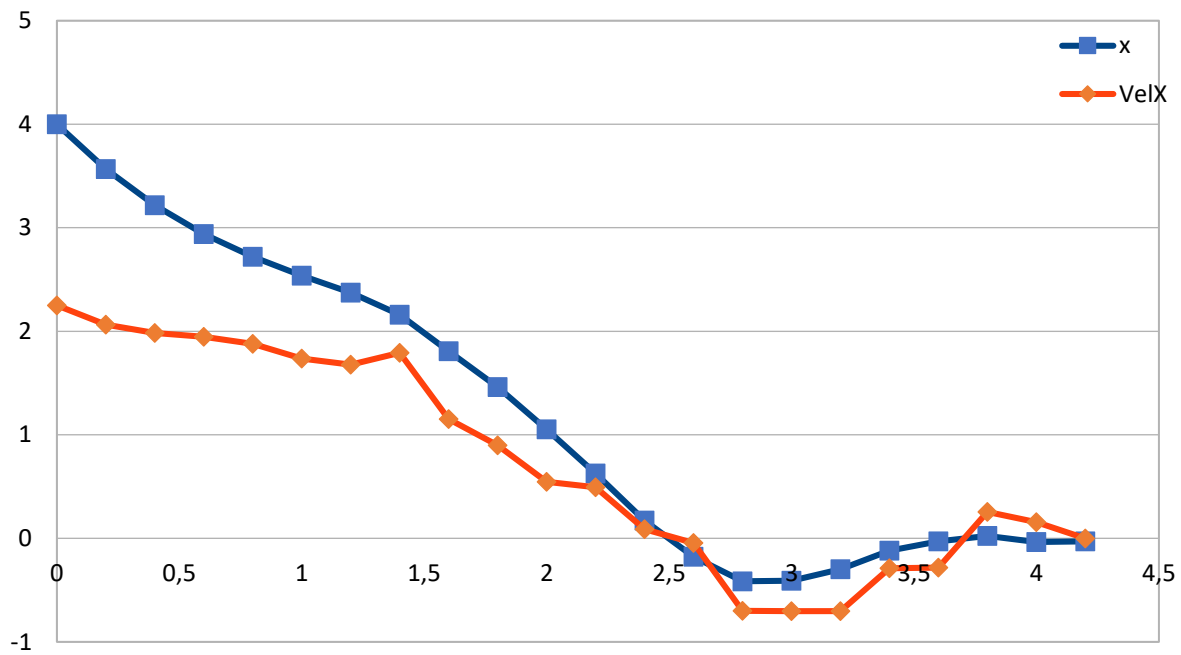


Fig. 4.67 Gráfico comparativo de X y VelX en presencia de alfa y beta

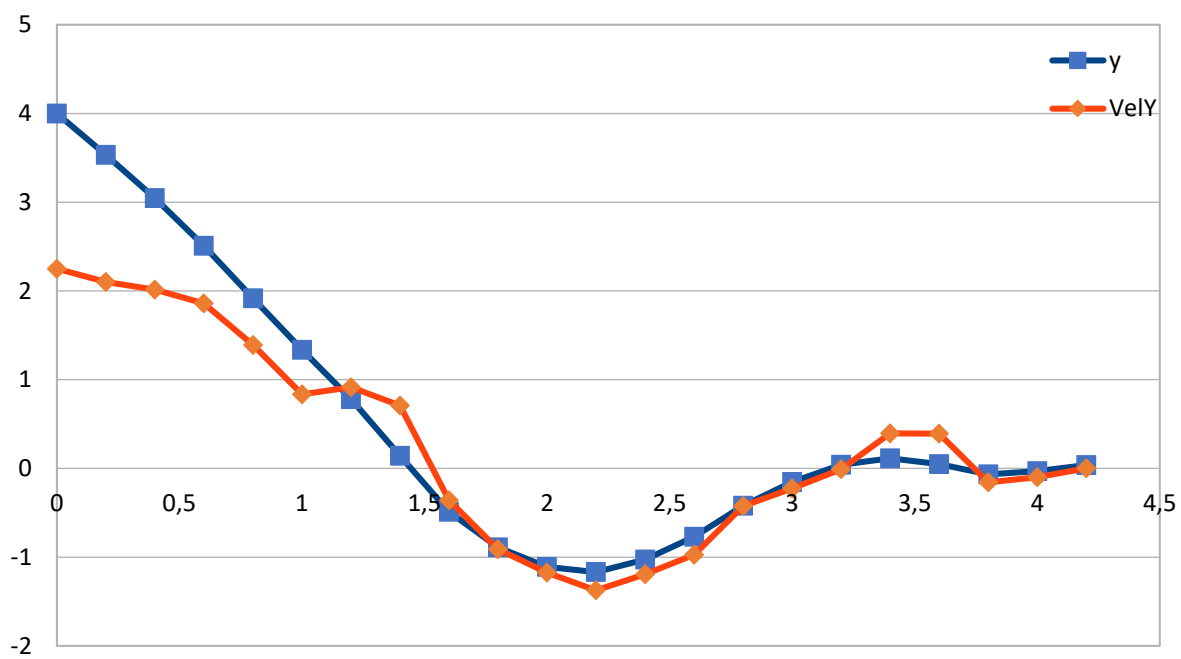


Fig. 4.68 Gráfico comparativo de Y y VelY en presencia de alfa y beta

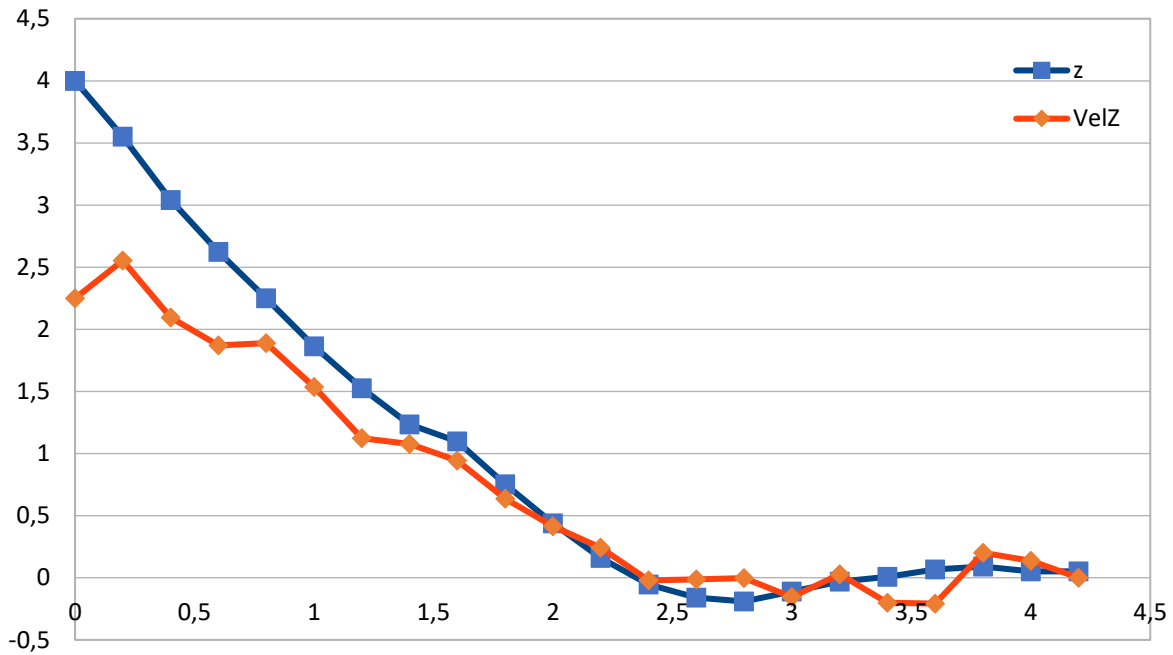


Fig. 4.69 Gráfico comparativo de Z y VelZ en presencia de alfa y beta

De los resultados presentados se deduce que en esta prueba se aprecia una ligera oscilación en el desplazamiento. La constante variación angular bidireccional hace que el sistema en el eje X y en el eje Y oscile excesivamente, además hay varios puntos en el eje Z (desde  $t=2,4$  hasta  $t=3,2$ ) donde la coordenada se hace inferior a 0. Este hecho puede suponer un problema porque significaría que si el dron se encuentra demasiado cerca en las coordenadas X e Y de la plataforma (como en este caso) podría haberse dado una colisión y que por tanto la región difusa de rectificación no ha sido suficientemente eficaz en presencia de la inclinación  $\alpha$  y  $\beta$  o que el área de parada (a 6cm en todas las direcciones teniendo en cuenta el sistema de referencia exento de rotación) no ha sido suficientemente grande como para parar el impacto. Se puede concluir por tanto que esta última prueba no ha superado todos los requisitos y que es necesario mejorar el controlador antes de probarlo en un sistema físico real

## CAPÍTULO 5. CONCLUSIONES Y TRABAJOS FUTUROS

### 5.1 Conclusiones:

En los experimentos realizados se ha podido constatar que el controlador de lógica difusa es fácil de comprender e implementar. Permite ajustar los parámetros sin demasiada información acerca de las condiciones del problema, ya que sólo necesita saber el punto de partida, la velocidad máxima y cuántas reglas o en cuántas partes se quiere separar el espacio de variables.

Además, el caso estudiado para un tipo concreto de dron es fácilmente generalizable a cualquier otro tipo. También, la disponibilidad de un sistema de visualización adecuado permitiría modificar la posición o la inclinación de la plataforma a tiempo real sin que ello influyera negativamente en el proceso de aterrizaje y sin que fuera necesario realizar cambios sustanciales en las funciones de membresía o en las reglas difusas.

En la mayoría de los estudios sobre aterrizaje implementando controladores basados en lógica difusa se aplican 4 o 5 reglas y generalmente las funciones de membresía se presentan con disposición trapezoidal. En este estudio se ha podido demostrar que existen combinaciones más eficaces, en el caso de los ejes Z y X cuyo espectro de valores oscila entre 0 y 4m para las distancias y entre 0 y 3m/s para velocidades la combinación que mejor responde es trapecio-triángulo con 4 reglas, en las simulaciones de la Y en la que también existen valores negativos, la combinación trapecio-trapecio con 5 reglas es la más adecuada.

En el caso de los ángulos,  $\alpha$  que presenta valores de entrada que comprenden de  $0^\circ$  a  $45^\circ$  y de  $135^\circ$  a  $180^\circ$  y de salida de  $-0,5\text{rad/s}$  hasta  $0,5\text{rad/s}$ , muestra mejores resultados con una combinación trapecio-triángulo y 5 reglas mientras que  $\beta$  cuyos valores de inclinación son de  $0^\circ$  a  $45^\circ$  y una velocidad de giro que toma valores de 0 a 0,5 presenta sus mejores resultados con una combinación triángulo-triángulo y 4 reglas.

En las pruebas que se han realizado comparando las diferentes funciones de membresía, se ha logrado que el dron pare en un área, definida como área de parada, a 2cm de distancia en todos los ejes y con un error de inclinación de  $\pm 0,5^\circ$  respecto a la inclinación de la plataforma. Posteriormente, se ha evaluado la respuesta de todo el sistema en conjunto y debido a la complejidad de los nuevos escenarios ha sido necesario el aumento del área de parada de 2cm a 6cm.

Los resultados han sido satisfactorios en ausencia de inclinación o cuando se presenta inclinación respecto a uno de los ejes. Los tiempos de aterrizaje han sido adecuados, la aproximación ha sido gradual y las variaciones de velocidad casos no han impedido un correcto aterrizaje.

Sin embargo, debido a la dificultad del último escenario en el que se presenta la plataforma con inclinación  $\alpha$  y  $\beta$ , no se han podido encontrar unos parámetros que se ajustaran suficientemente bien al problema planteado, y ni una región de rectificación en las funciones de membresía implementada en este último escenario, ni un aumento considerable del área de parada han impedido que hasta en 4 ocasiones la lectura teórica del eje Z estuviera por debajo de 0, implicando así una posible colisión.

La problemática es más complicada de lo que pueda parecer, si se aumenta la región de rectificación se corre el riesgo de que el dron nunca llegue a acercarse lo suficiente oscilando entre un área de parada ya lejana y distancias superiores, y si se amplía el área de parada se corre el riesgo de que el dron sufra daños físicos por haber detenido los motores cuando no estaba lo suficientemente cerca.

## **5.2 Trabajos futuros:**

A lo largo de este proyecto se han mencionado algunas de las posibles mejoras que se pueden realizar para confeccionar un sistema de control completo y funcional. En primer lugar, es necesaria la creación de un sistema de visualización capaz de determinar la inclinación de la plataforma, recomendable, que siga un funcionamiento similar al paquete mencionado en este proyecto llamado `Ar_track_alvar`.

En segundo lugar, se propone la creación de un sistema de optimización para el sistema de control de lógica difusa, basado en algoritmos genéticos. Esta posibilidad ya se ha barajado en algunos de los estudios utilizados como bibliografía [5].

En tercer lugar, se propone el uso de simuladores 3D como Gazebo o V-rep para visualizar gráficamente las pruebas que se quieran realizar con el fin de comprender y analizar mejor las trayectorias que ha podido tomar el vehículo.

Por último, se propone que el dron busque la plataforma de no tenerla en el campo visual y modifique

su trayectoria, velocidad o rotación si la plataforma fuera móvil, aunque como se ha anticipado este último punto no supondría grandes variaciones en el controlador debido a su versatilidad.

## BIBLIOGRAFÍA

- [1] Los 14 usos de los drones que no conocías, *DonWeb*, 2014. [En línea]. Disponible en: <http://agencia.donweb.com/los-14-usos-de-drones-que-seguro-no-conocias/>. (Acceso: septiembre 2018)
- [2] MercadoLibre Perú S.R.L, “Drone Ala Fija Survey Scout Para Topografia”, *Mercado libre*, [En línea]. Disponible en:  
[https://articulo.mercadolibre.com.pe/MPE-421960950-drone-ala-fija-survey-scout-para-topografia-\\_JM](https://articulo.mercadolibre.com.pe/MPE-421960950-drone-ala-fija-survey-scout-para-topografia-_JM) (Acceso: agosto 2018)
- [3] C. Otero, “Por qué este dron vuela como un murciélago es un triunfo de la ciencia”, *BeTech*, 02-02-2017 [En línea]. Disponible en: [https://as.com/betech/2017/02/02/portada/1486043855\\_246179.html](https://as.com/betech/2017/02/02/portada/1486043855_246179.html) (Acceso: septiembre 2018)
- [4] DroneShop, “Los drones y sus diferentes plataformas”, *Drones.mx*, 02-05-2016 [En línea]. Disponible en: <https://drones.mx/blog/los-drones-diferentes-plataformas/> (Acceso: agosto 2018)
- [5] A. Radwan Ibrahim, "Design and Test a Control System for a Quadcopter." Bachelor Thesis, Mechatronics Department, German University in Cairo, Cairo, Egypt, 2016.
- [6] A. Vázquez Ruano, “Control system for autonomous landing on moving platforms”, Trabajo de fin de grado, Laboratorio de sistemas inteligentes, Universidad Carlos III de Madrid, Madrid, España, 2017.
- [7] “Low priced RC Hexacopter RC Hexacopter | MJX x600 RC Quadcopter”, *Drone Flyers*, [En línea]. Disponible en: <https://droneflyers.wordpress.com/low-priced-rc-hexacopter/> (Acceso: septiembre 2018)
- [8] E. Santana, “Tipos de drones – Conoce todos los tipos de drones que existen | Clasificación de drones | Categorías de drones”, *XDrones*, [En línea]. Disponible en: <http://www.xdrones.es/tipos-de-drones-clasificacion-de-drones-categorias-de-drones/> (Acceso: septiembre 2018)
- [9] P. Benavidez, J. Lambert, A. Jaimes y M. Jamshidi “Landing of an Ardrone 2.0 quadcopter on a mobile base using fuzzy logic”, *International Journal of Complex Systems*, vol. 1, No. 1-2, pp. 5-25, 2013. [En línea]. Disponible en: <http://ij-complexsys.org/>

- [10] C. Hajiyeu., H.E Soken y S.Y. Vural, “Classic Controller Design for Unmanned Aerial Vehicles”. en *State Estimation and Control for Low-cost Unmanned Aerial Vehicles*, 1ª. Suiza: Springer International, 2015, 141-170.
- [11] C. Hajiyeu., H.E Soken y S.Y. Vural, “Linear Quadratic Regulator Controller Design”. en *State Estimation and Control for Low-cost Unmanned Aerial Vehicles*, 1ª. Suiza: Springer International, 2015, 171-200.
- [12] P. Górecki, L. Caponetti y C. Castiello “Fuzzy Techniques for Text Localisation in Images”, *Computational Intelligence in Multimedia Processing: Recent Advances*, 1, Heidelberg: Springer-Verlag Berlin Heidelberg, 2008, (pp.233-270) [En línea] Disponible en: [https://www.researchgate.net/figure/An-example-of-a-crisp-set-and-a-fuzzy-set-with-Gaussian-membership-function\\_fig1\\_225357187](https://www.researchgate.net/figure/An-example-of-a-crisp-set-and-a-fuzzy-set-with-Gaussian-membership-function_fig1_225357187) (Acceso: septiembre 2018)
- [13] C. Hajiyeu., H.E Soken y S.Y. Vural, “Fuzzy Logic-Based Controller Design”, en *State Estimation and Control for Low-cost Unmanned Aerial Vehicles*, 1ª. Suiza: Springer International, 2015, 201-221.
- [14] E. Koslosky, A.S. de Oliveira, M.A. Wehrmeister y J.A. Fabro, “Designing Fuzzy Logic Controllers for ROS-Based Multirotors”, en *Robot Operating System (ROS)*. Anis Koubaa, Cham: Springer, 2017, 41-82.
- [15] S. Bagwe, “Question: Explain any four defuzzification with suitable example”, *Ques10*, 05-2016, [En línea]. Disponible en: <http://www.ques10.com/p/13349/explain-any-four-defuzzification-with-suitable-e-1/> (Acceso: septiembre 2018)
- [16] A. Karabegovic, Z. Avdagic y M. Ponjavic “Applications of Fuzzy Logic in Geographic Information Systems for Multiple Criteria Decision Making”, *GAUSS Centar za geoprostorna istraživanja Sarajevo*. [En línea]. Disponible en: <http://www.gis.ba/en/applications-of-fuzzy-logic-in-geographic-information-systems-for-multiple-criteria-decision-making/> (Acceso: septiembre 2018)
- [17] A. Martínez Romero, “ROS/Concepts”, *ROS*, 2014-06-21, [En línea]. Disponible en: <http://wiki.ros.org/ROS/Concepts> (Acceso: septiembre 2018)
- [18] I. Saito, “Ar\_track\_alvar”, *ROS*, [En línea]. Disponible en: [http://wiki.ros.org/ar\\_track\\_alvar](http://wiki.ros.org/ar_track_alvar) (Acceso: septiembre 2018)
- [19] J. Lim, “PX4 Offroad Control Using MAVROS on ROS”, *404warehouse*, 20-12-2015, [En línea]. Disponible en: <https://404warehouse.net/2015/12/20/autopilot-offboard-control-using-mavros-package-on-ros/> (Acceso: agosto 2018)



- [20] MAVROS, *Droncode*, [En línea]. Disponible en: [https://dev.px4.io/en/ros/mavros\\_installation.html](https://dev.px4.io/en/ros/mavros_installation.html) (Acceso: agosto 2018)
- [21] E. Santana, “MAVLink: protocolo de comunicación para drones”, *Xdrones*, [En línea]. Disponible en: <http://www.xdrones.es/mavlink/> (Acceso: septiembre 2018)
- [22] L. Meier, “MAVLink”, *Droncode*, 2009, [En línea]. Disponible en: <https://mavlink.io/en/> (Acceso: septiembre 2018)
- [23] PX4 Dev Team, “Pixhawk Series”, *Droncode*, 03-09-2018. [En línea]. Disponible en: [https://docs.px4.io/en/flight\\_controller/pixhawk\\_series.html](https://docs.px4.io/en/flight_controller/pixhawk_series.html) (Acceso: agosto 2018)
- [24] PX4 Dev Team, “Basic Concepts”, *Droncode*, 03-09-2018. [En línea]. Disponible en: [https://docs.px4.io/en/getting\\_started/px4\\_basic\\_concepts.html#vehicle\\_controller](https://docs.px4.io/en/getting_started/px4_basic_concepts.html#vehicle_controller) (Acceso: agosto 2018)
- [25] PX4 Dev Team, “Sensors”, *Droncode*, 03-09-2018. [En línea]. Disponible en: [https://docs.px4.io/en/getting\\_started/sensor\\_selection.html](https://docs.px4.io/en/getting_started/sensor_selection.html) (Acceso: agosto 2018)
- [26] PX4 Dev Team, “Peripherals”, *Droncode*, 03-09-2018. [En línea]. Disponible en: <https://docs.px4.io/en/peripherals/> (Acceso: agosto 2018)
- [27] PX4 Dev Team, “Pixhawk Wiring Quick Start”, *Droncode*, 03-09-2018. [En línea]. Disponible en: [https://docs.px4.io/en/assembly/quick\\_start\\_pixhawk.html](https://docs.px4.io/en/assembly/quick_start_pixhawk.html) (Acceso: agosto 2018)
- [28] PX4 Dev Team, “Pixhawk Wiring Quick Start”, *Droncode*, 03-09-2018. [En línea]. Disponible en: [https://docs.px4.io/en/assembly/quick\\_start\\_pixhawk.html](https://docs.px4.io/en/assembly/quick_start_pixhawk.html) (Acceso: agosto 2018)
- [29] PX4 Dev Team, “Flight Controller/Sensor Orientation”, *Droncode*, 03-09-2018. [En línea]. Disponible en: [https://docs.px4.io/en/config/flight\\_controller\\_orientation.html](https://docs.px4.io/en/config/flight_controller_orientation.html) (Acceso: agosto 2018)

